

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Ульяновский государственный университет»
Факультет математики и информационных технологий
Кафедра информационной безопасности и теории управления

С. М. Рацев

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ
ПО ПРОГРАММИРОВАНИЮ**

Ульяновск
2014

УДК 004.4
ББК 32.965
Р 27

*Печатается по решению Ученого совета
факультета математики и информационных технологий
Ульяновского государственного университета
(протокол № 2/14 от 15.04.2014)*

Рецензент – кандидат технических наук, доцент *Е.Г. Чекал*.

Рацеев С. М.
Р 27 **Лабораторный практикум по программированию** : методические указания / С. М. Рацеев. – Ульяновск : УлГУ, 2014. – 92 с.

В данном лабораторном практикуме по программированию приводятся варианты индивидуальных заданий по таким темам, как условные операторы, циклы, одномерные массивы, матрицы, строки, структуры данных, текстовые и двоичные файлы, линейные динамические структуры данных, бинарные деревья, биты памяти.

Предназначено для студентов младших курсов, изучающих язык программирования «Си».

УДК 004.4
ББК 32.965

© Рацеев С.М., 2014
© Ульяновский государственный университет, 2014

Оглавление

Требования к программам	4
Лабораторная работа № 1. Программирование циклических вычислительных процессов.....	6
Некоторые стандартные математические функции.....	10
Лабораторная работа № 2. Проверка попадания точки в область на плоскости.....	12
Лабораторная работа № 3. Числовые и функциональные ряды	16
Лабораторная работа № 4. Вычисление характеристик числовых последовательностей.....	20
Лабораторная работа № 5. Определение характеристик одномерных массивов	24
Лабораторная работа № 6. Модификация одномерных массивов	30
Лабораторная работа № 7. Обработка текстовых строк	36
Лабораторная работа № 8. Обработка и модификация двумерных массивов	42
Лабораторная работа № 9. Рекурсия	49
Лабораторная работа № 10. Текстовые файлы.....	57
Лабораторная работа № 11. Двоичные файлы. Базы данных.....	62
Лабораторная работа № 12. Линейные списки	82
Лабораторная работа № 13. Обработка линейных динамических структур.....	86
Лабораторная работа № 14. Обработка бинарных деревьев.....	87
Лабораторная работа № 15. Работа с битами памяти.....	90
Литература	91

Требования к программам

В каждой лабораторной работе задания разделены на три уровня сложности:

- уровень А (начальный);
- уровень В (средний и выше среднего);
- уровень С (продвинутый).

Необходимо в каждой лабораторной работе выбрать уровень сложности по своим силам и в соответствующем уровне выбрать один вариант для выполнения задания.

При этом следует учесть следующие моменты. Чтобы на экзамене рассчитывать на оценку **«отлично»**, помимо *соответствующего* ответа на экзаменационные вопросы, необходимо:

- отсутствие долгов по лабораторным работам к моменту проведения экзамена;
- своевременная сдача лабораторных работ;
- выполнение не менее 50 % лабораторных работ с уровнем сложности С и ни одной лабораторной работы с уровнем сложности А;
- решение дополнительных задач соответствующего уровня.

Чтобы на экзамене рассчитывать на оценку **«хорошо»**, помимо *соответствующего* ответа на экзаменационные вопросы, необходимо:

- отсутствие долгов по лабораторным работам к моменту проведения экзамена;
- своевременная сдача лабораторных работ;
- выполнение не менее 75 % лабораторных работ с уровнем сложности В;
- решение дополнительных задач соответствующего уровня.

Важно!!! Не стоит сдавать лабораторные работы с уровнем сложности В или С, если программы были написаны кем-то другим. Такие работы приниматься не будут.

При выполнении заданий с уровнем сложности С *рекомендуется* использовать эффективные алгоритмы для решения соответствующих задач и производить вычисления сложности алгоритмов.

Сроки сдачи лабораторных работ

Для групп КБ, ИС, ИТСС:

<i>Номер лабораторной работы</i>	<i>Срок сдачи</i>
1, 2	сентябрь
3, 4	октябрь
5, 6, 7, 8	ноябрь
9, 10, 11, 12	декабрь

Для групп МОАИС, САиУ, АТПШ, Ав:

<i>Номер лабораторной работы</i>	<i>Срок сдачи</i>
1	сентябрь
2, 3	октябрь
4, 5	ноябрь
6, 7, 8	декабрь
9	февраль
10, 11	март
12, 13	апрель
14, 15	май

Программа должна иметь структурированный вид. Это значит, что операторы надо располагать с отступами от левого края согласно принципу подчиненности операторов друг другу. В результате, например, открывающейся операторной скобке { будет соответствовать закрывающая скобка }, расположенная либо в той же строке, либо на одной из нижних строк, но точно под скобкой {. Например,

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 0; i < 10; i++)
        printf("%d\n", i);
    return 0;
}
```

Лабораторная работа № 1. Программирование циклических вычислительных процессов

Для функции $f(x) = xe^x + 2\sin x - \sqrt{|x^3 - x^2|}$ составить программу построения таблицы значений функции при изменении аргумента от $A = -5$ до $B = 5$ с шагом $H = 0,1$. Значения A , B и H объявить как константы в директивах `#define`. В каждой строке выводить значение аргумента и соответствующее ему значение функции в форматированном виде с 4 знаками после запятой. Кроме того, в конце таблицы нужно напечатать отдельной строкой значение, которое требуется вычислить в соответствии с индивидуальным вариантом.

Уровень А

Варианты заданий

- A1. Среднее арифметическое всех отрицательных значений функции.
- A2. Сумма таких значений функции, которые находятся на отрезке от M до N .
- A3. Среднее арифметическое таких значений функции, которые находятся на отрезке от M до N .
- A4. Среднее арифметическое таких значений функции, которые больше числа M .
- A5. Среднее арифметическое всех неотрицательных значений функции.
- A6. Количество всех отрицательных значений функции.
- A7. Количество таких значений функции, которые находятся на отрезке от M до N .
- A8. Количество всех неотрицательных значений функции.
- A9. Среднее арифметическое таких значений функции, которые меньше числа M .
- A10. Сумма таких значений функции $y=f(x)$, для которых выполнено неравенство $x > f(x)$.
- A11. Количество таких значений функции, которые меньше числа M .

A12. Среднее арифметическое таких значений функции, которые меньше числа М.

A13. Сумма таких значений функции, которые не принадлежат отрезку от М до N.

A14. Количество таких значений функции $y=f(x)$, для которых выполнено неравенство $x < f(x)$.

A15. Количество таких значений функции $y=f(x)$, для которых выполнено равенство $|x| = |f(x)|$.

A16. Сумма таких значений функции $y=f(x)$, для которых выполнено неравенство $x^2 > f(x)$.

A17. Сумма таких значений функции, которые по модулю больше числа М.

Пример программы уровня А. Найти среднее арифметическое таких значений функции $y=\sin(x)+x^2$, для которых выполнено неравенство $f(x) > x^2$.

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#define A -3.0
```

```
#define B 3.0
```

```
#define H 0.2
```

```
/* функция F() вычисляет значение функции  $y = \sin(x)+x^2$  в точке x */
```

```
double F(double x)
```

```
{
```

```
    return sin(x) + x*x;
```

```
}
```

```
int main( )
```

```
{
```

```
    double x, y, sum;
```

```
    int count;
```

```
    sum = count = 0;
```

```
    for(x = A; x <= B; x += H)
```

```

{
    y = F(x);
    printf("%10.4f %10.4f\n", x, y);
    if (y > x*x)
    {
        count++;
        sum += y;
    }
}
if (count == 0)
    puts("count = 0");
else printf("sred = %10.4f\n", sum/count);
return 0;
}

```

Уровень В

Варианты заданий

- В1. Количество отрицательных значений функции, имеющих нечетную целую часть.
- В2. Среднее арифметическое таких значений функции, которые имеют в младшем разряде целой части цифру, большую 3.
- В3. Среднее арифметическое таких значений функции, которые имеют дробную часть, меньшую 0,5.
- В4. Количество значений функции, имеющих в младшем разряде целой части цифру, большую 3.
- В5. Среднее арифметическое таких значений функции, которые имеют четную целую часть.
- В6. Количество неотрицательных значений функции, имеющих четную целую часть.
- В7. Сумма таких значений функции, которые имеют дробную часть, большую 0,5.
- В8. Сумма таких значений функции, которые имеют целую часть, кратную Z.

В9. Среднее арифметическое таких значений функции, которые имеют нечетную целую часть.

В10. Количество положительных значений функции, которые имеют четную целую часть и дробную часть, меньшую 0,5.

В11. Сумма неотрицательных значений функции, которые имеют нечетную целую часть и дробную часть, большую 0,2.

В12. Среднее арифметическое таких значений функции, которые имеют нечетную целую часть и дробную часть, большую 0,5.

В13. Сумма таких значений функции, целая часть которых двузначна.

В14. Количество таких значений функции, целая часть которых по модулю меньше числа M .

В15. Сумма таких значений функции, дробная часть которых больше 0,3 и меньше 0,7.

В16. Среднее арифметическое таких значений функции, целая часть которых по модулю больше числа M .

В17. Среднее арифметическое таких значений функции, дробная часть которых больше 0,2 и меньше 0,8.

Уровень С

Для решения следующих заданий разрешается использовать цикл только один раз.

Варианты заданий

С1. Минимальное среди неотрицательных значений функции и количество таких минимальных значений.

С2. Максимальное среди значений функции, имеющих целую часть, кратную Z , и количество таких максимальных значений.

С3. Минимальное среди значений функции, имеющих дробную часть, меньшую 0,5, и количество таких минимальных значений.

С4. Минимальное среди значений функции, имеющих четную целую часть, и количество таких минимальных значений.

C5. Максимальное среди значений функции, имеющих нечетную целую часть, и количество таких максимальных значений.

C6. Максимальное среди значений функции, имеющих в младшем разряде целой части цифру, меньшую 4, и количество таких максимальных значений.

C7. Максимальное среди значений функции, имеющих дробную часть больше 0,4, и количество таких максимальных значений.

C8. Минимальное среди значений функции, которые имеют в младшем разряде целой части цифру, большую 3, и количество таких минимальных значений.

C9. Минимальное среди значений функции, целая часть которых меньше числа M , и количество таких минимальных значений.

C10. Максимальное среди значений функции, имеющих в младшем разряде целой части цифру, большую 2, и количество таких максимальных значений.

Некоторые стандартные математические функции

Стандартные математические функции находятся в файле *math.h*.

abs – абсолютное значение целого числа – $|x|$

```
int abs(int x);
```

labs – абсолютное значение «длинного» целого числа – $|x|$:

```
long labs(long x);
```

fabs – абсолютное значение числа с плавающей точкой – $|x|$:

```
double fabs(double x);
```

sqrt – извлечение квадратного корня:

```
double sqrt(double x);
```

pow – возведение в степень:

```
double pow(double x, double y);
```

cos – косинус – $\cos x$ (здесь и далее x задается в радианах):

```
double cos(double x);
```

sin – синус – $\sin x$:

```
double sin(double x);
```

tan – тангенс – ***tg x***:

double tan(double x);

acos – арккосинус – ***arccos x***:

double acos(double x);

asin – арксинус – ***arcsin x***:

double asin(double x);

atan – арктангенс – ***arctg x***:

double atan(double x);

atan2 – арктангенс – ***arctg x/y***:

double atan2(double x, double y);

exp – экспонента:

double exp(double x);

log – натуральный логарифм – ***ln x***:

double log(double x);

log10 – десятичный логарифм – ***log₁₀x***:

double log10(double x).

Лабораторная работа № 2. Проверка попадания точки в область на плоскости

Для функции из лабораторной работы № 1 составить программу построения таблицы значений функции, состоящую из n строк при изменении аргумента от A до B .

В каждой строке таблицы выводить значение аргумента x и соответствующее ему значение функции y в форматированном виде с 4 знаками после запятой. В конце таблицы вывести отдельной строкой среднее арифметическое всех значений функции.

Кроме того, программа должна определить, сколько точек с координатами (x, y) из полученной таблицы попадает внутрь области, ограниченной фигурой, соответствующей индивидуальному заданию.

Уровень А

В качестве области выступает прямоугольник со следующими вершинами.

Варианты заданий

A1. (-9, -10), (-9, 2), (2, 2), (2, -10).

A2. (-6, -7), (-6, 3), (7, 3), (7, -7).

A3. (-8, -6), (-8, 4), (6, 4), (6, -6).

A4. (-10, -9), (-10, 3), (2, 3), (2, -9).

A5. (-7, -8), (-7, 6), (4, 6), (4, -8).

A6. (-10, -6), (-10, 7), (0, 7), (0, -6).

A7. (-6, -10), (-6, 4), (5, 4), (5, -10).

A8. (-9, -10), (-9, 3), (5, 3), (5, -10).

A9. (-7, -6), (-7, 5), (3, 5), (3, -6).

A10. (-9, -7), (-9, 3), (5, 3), (5, -7).

A11. (-10, -7), (-10, 6), (2, 6), (2, -7).

A12. (-10, -6), (-10, 4), (0, 4), (0, -6).

A13. (-10, -7), (-10, 6), (2, 6), (2, -7).

A14. (-9, -9), (-9, 2), (2, 2), (2, -9).

A15. (-9, -6), (-9, 8), (3, 8), (3, -6).

A16. (-9, -6), (-9, 5), (2, 5), (2, -6).

A17. (-6, -10), (-6, 3), (4, 3), (4, -10).

Пример программы уровня А. Найти количество точек (x, y) , где $y = \sin(x) + x^2$, которые попадают внутрь области, ограниченной прямоугольником с вершинами $(-1, -2)$, $(-1, 3)$, $(5, 3)$, $(5, -2)$.

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#define A -3.0
```

```
#define B 3.0
```

```
double F(double x)
```

```
{
```

```
    return sin(x) + x*x;
```

```
}
```

```
int main( )
```

```
{
```

```
    double x, y, h;
```

```
    int count, n;
```

```
    count = 0;
```

```
    printf("n = "); scanf("%d", &n); /* количество строк в таблице */
```

```
    h = (B - A) / (n - 1);
```

```
    for(x = A; x <= B; x += h)
```

```
    {
```

```
        y = F(x);
```

```
        printf("%10.4f %10.4f\n", x, y);
```

```
        if (x >= -1 && x <= 5 && y >= -2 && y <= 3)
```

```
            count++;
```

```
    }
```

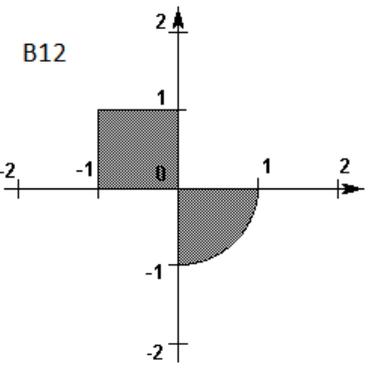
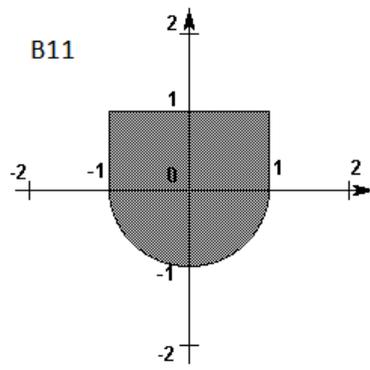
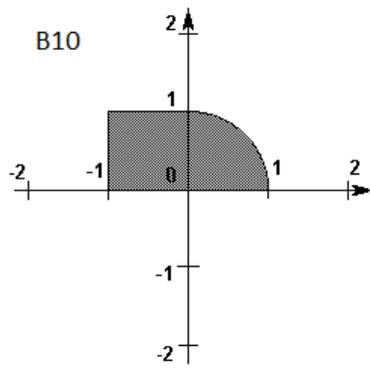
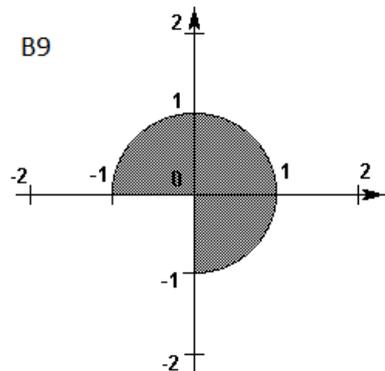
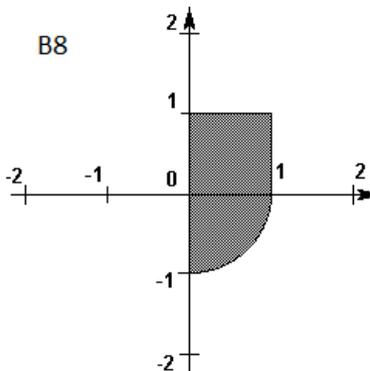
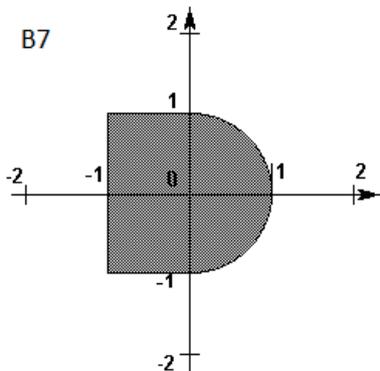
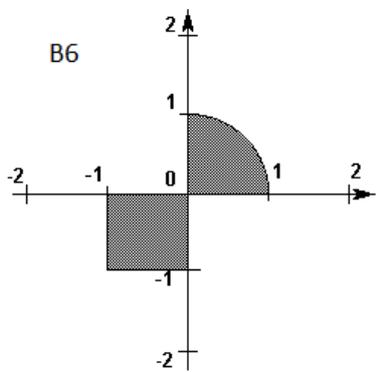
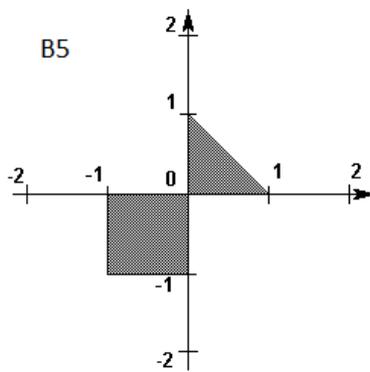
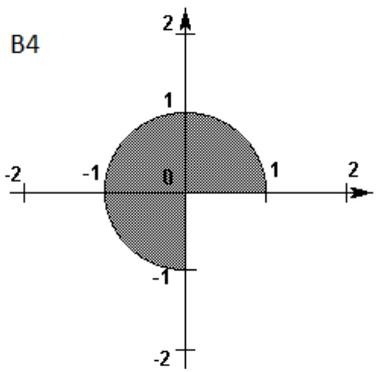
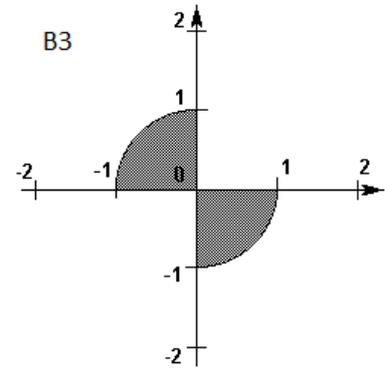
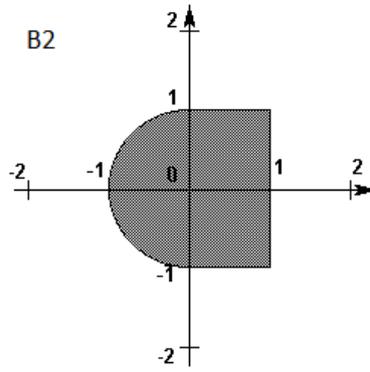
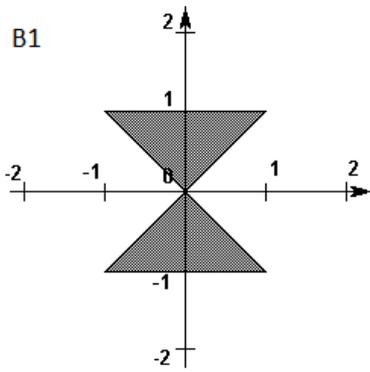
```
    printf("count = %d\n", count);
```

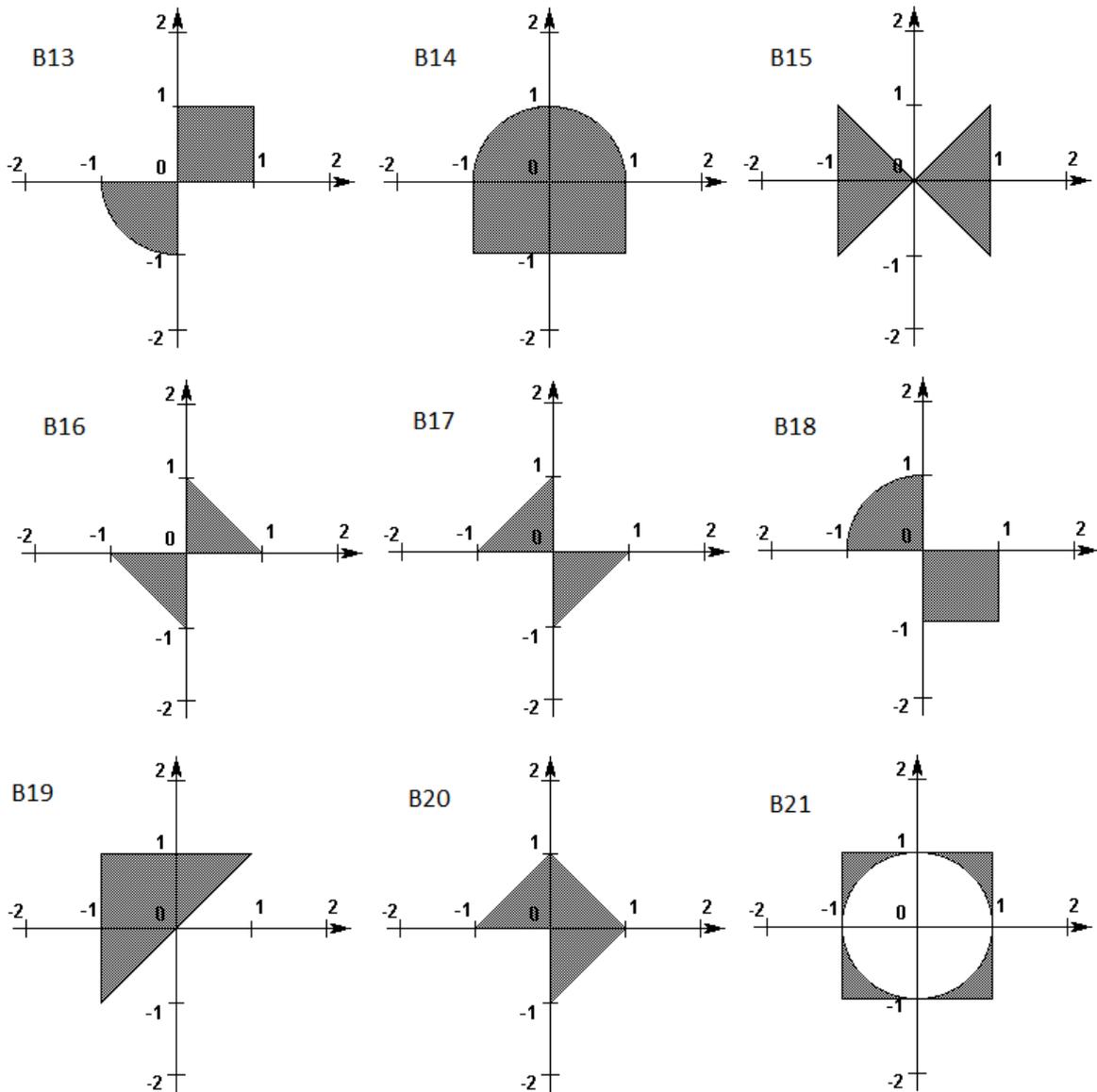
```
    return 0;
```

```
}
```

Уровень В

Варианты заданий





Уровень С

Сгенерировать четыре случайные точки плоскости (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) с тем условием, что точка (x_1, y_1) принадлежит первой четверти координатной плоскости, (x_2, y_2) – второй четверти, (x_3, y_3) – третьей четверти, (x_4, y_4) – четвертой четверти. После этого найти количество точек (x, y) из таблицы, принадлежащих полученной области. Сделать это двумя способами – через уравнения прямых и через площади треугольников.

Лабораторная работа № 3. Числовые и функциональные ряды

Уровень А

Для бесконечного числового ряда

$$a_1 + a_2 + \dots + a_n + \dots = \sum_{n=1}^{\infty} a_n,$$

члены которого вычисляются по формуле, соответствующей индивидуальному варианту, вычислить сумму первых n членов ряда с использованием функции

double Sum(int n).

Варианты заданий

$$A1. a_n = \frac{1}{n^2 + 10} \quad A2. a_n = \frac{1}{n^2 + n} \quad A3. a_n = \frac{2}{n^2 + 3}$$

$$A4. a_n = \frac{2}{n^2 + 1} \quad A5. a_n = \frac{3}{n^2 + 2n} \quad A6. a_n = \frac{1}{n^2 + 2n + 1}$$

$$A7. a_n = \frac{n}{n^3 + 1} \quad A8. a_n = \frac{n}{n^3 + n + 1} \quad A9. a_n = \frac{n - 1}{n^3 + 3}$$

$$A10. a_n = \frac{n + 1}{n^3 + 1} \quad A11. a_n = \frac{n + 1}{n^3 + 2n} \quad A12. a_n = \frac{n + 2}{n^3 + 3}$$

$$A13. a_n = \frac{n - 3}{n^3 - n + 1} \quad A14. a_n = \frac{n + 10}{n^3 + 5n + 1} \quad A15. a_n = \frac{2}{n^3 + n}$$

$$A16. a_n = \frac{3}{n^2 - n + 1} \quad A17. a_n = \frac{10}{n^2 + 5n + 1} \quad A18. a_n = \frac{1}{n^3 - 3n}$$

Пример программы уровня А. Вычислить сумму первых n членов числового ряда, члены которого вычисляются по формуле

$$a_n = \frac{5}{n^2 + 10n - 1}.$$

```
#include <stdio.h>

double Sum(int n)
{
    double sum = 0;
    int i;
    for(i = 1; i <= n; i++)
        sum += 5.0 / (i*i + 10*i - 1);
    return sum;
}

int main( )
{
    int n;
    printf("n = "); scanf("%d", &n);
    printf("sum = %.10f\n", Sum(n));
    return 0;
}
```

Уровень В

Для бесконечного числового ряда

$$a_1 + a_2 + \dots + a_n + \dots = \sum_{n=1}^{\infty} a_n,$$

члены которого вычисляются по формуле, соответствующей индивидуальному варианту, вычислить приближенную сумму двумя способами: сумму первых n членов ряда и сумму с точностью до eps (эпсилон) в двух разных функциях:

```
double Sum1(int n);
double Sum2(double eps).
```

Запрещается использование функции `pow()` и условных операторов «if» и «?:».

Варианты заданий

$$B1. a_n = (-1)^n \frac{n+1}{n^3+1} \quad B2. a_n = (-1)^n \frac{1}{n^2+1}$$

$$B3. a_n = (-1)^n \frac{n}{n^3-n} \quad B4. a_n = (-1)^n \frac{1}{n^2+n+2}$$

$$B5. a_n = (-1)^n \frac{n-1}{n^4-n} \quad B6. a_n = (-1)^n \frac{n-1}{n^3+n}$$

$$B7. a_n = (-1)^n \frac{n-1}{n^3+5n} \quad B8. a_n = (-1)^n \frac{1}{n^2+n-1}$$

$$B9. a_n = (-1)^n \frac{n}{n^3-n+2} \quad B10. a_n = (-1)^n \frac{n+2}{n^3+n-2}$$

$$B11. a_n = (-1)^n \frac{1}{n^2-n+2} \quad B12. a_n = (-1)^n \frac{2}{n^2+n+3}$$

$$B13. a_n = (-1)^n \frac{1}{n^4+10} \quad B14. a_n = (-1)^n \frac{1}{n^3+n}$$

$$B15. a_n = (-1)^n \frac{n}{n^4+1} \quad B16. a_n = (-1)^n \frac{n}{n^3-n+1}$$

$$B17. a_n = (-1)^n \frac{1}{n^3+10n} \quad B18. a_n = (-1)^n \frac{n}{n^3+n+1}$$

Уровень С

Для функционального ряда

$$a_1x + a_2x^2 + \dots + a_nx^n + \dots = \sum_{n=1}^{\infty} a_nx^n,$$

коэффициенты которого вычисляются по формуле, соответствующей индивидуальному варианту, вычислить приближенную сумму при фиксированном значении x двумя способами: сумму первых n членов ряда и сумму с точностью до ϵ (эпсилон) в двух разных функциях:

`double Sum1(double x, int n);`

`double Sum2(double x, double eps).`

Запрещается использование условных операторов `if` и «`? :`», функции `pow()`, вторичное вычисление факториала и возведение в степень на каждом шаге итерации (использовать рекуррентные соотношения $n! = n \cdot (n-1)!$, $a^n = a \cdot a^{n-1}$).

Варианты заданий

$$C1. a_n = (-1)^n \frac{n+1}{3^n \cdot n!} \quad C2. a_n = (-1)^n \frac{n}{2^n \cdot (n-1)!}$$

$$C3. a_n = (-1)^n \frac{2^n}{n!} \quad C4. a_n = (-1)^n \frac{n}{(n+1)!}$$

$$C5. a_n = (-1)^n \frac{3^n}{(n-1)!} \quad C6. a_n = (-1)^n \frac{n+1}{(n-1)!}$$

$$C7. a_n = (-1)^n \frac{n+1}{2^n \cdot n!} \quad C8. a_n = (-1)^n \frac{n}{3^n \cdot (n+1)!}$$

$$C9. a_n = (-1)^n \frac{3^n}{(n+2)!} \quad C10. a_n = (-1)^n \frac{2^n}{3^{n+1} \cdot (n+1)!}$$

Лабораторная работа № 4. Вычисление характеристик числовых последовательностей

Последовательность данных не всегда нужно сохранять в памяти. Поэтому нужно уметь обрабатывать последовательность по мере поступления ее элементов: при чтении файла, при вводе некоторых данных с клавиатуры и т.д.

Пусть имеется последовательность произвольных целых ненулевых чисел, завершающаяся числом 0 (*число 0 в последовательность не входит*). Количество элементов в последовательности заранее неизвестно, поэтому в данной работе *использовать массивы (статические и динамические) нельзя*.

Уровень А

Варианты заданий

- A1. Напечатать в обратном порядке наибольший элемент последовательности.
- A2. Найти количество наибольших и наименьших элементов последовательности.
- A3. Найти количество простых чисел в последовательности, больших заданного числа M .
- A4. Найти сумму таких элементов последовательности, которые больше всех предшествующих им элементов.
- A5. Найти количество четных чисел последовательности, сумма цифр в которых не превышает заданного числа P .
- A6. Найти количество таких элементов последовательности, которые превосходят сумму всех предшествующих элементов.
- A7. Напечатать в обратном порядке наименьший элемент последовательности.
- A8. Найти сумму таких элементов последовательности, которые имеют в младшем разряде цифру 7.
- A9. Найти сумму таких элементов последовательности, которые меньше всех предшествующих им элементов.

A10. Найти количество таких элементов последовательности, которые не превосходят сумму всех предшествующих элементов.

A11. Найти сумму нечетных чисел последовательности, сумма цифр в которых превышает заданное число P.

A12. Найти количество таких элементов последовательности, которые имеют в младшем разряде цифру 3.

A13. Найти количество двузначных элементов последовательности.

A14. Найти количество таких элементов последовательности, сумма цифр в которых является четным числом.

A15. Найти количество составных чисел в последовательности.

A16. Найти сумму нечетных чисел последовательности, количество цифр в которых не превышает заданного числа P.

A17. Найти количество таких элементов последовательности, которые имеют в своем составе цифру 5.

Пример программы уровня А. Найти количество таких элементов последовательности целых чисел, первая цифра которых равна 5.

```
#include <stdio.h>
```

```
#include <math.h>
```

```
/* проверка, является ли цифра d первой цифрой числа a */
```

```
int Check(int a, int d)
```

```
{
```

```
    int b = abs(a);
```

```
    while (b > 9)
```

```
        b /= 10;
```

```
    return b == d;
```

```
}
```

```
int main( )
```

```
{
```

```
    int a, count = 0;
```

```
    printf("a = "); scanf("%d", &a);
```

```
    while (a)
```

```

{
    if (Check(a, 5))
        count++;
    printf("a = "); scanf("%d", &a);
}
printf("count = %d\n", count);
return 0;
}

```

Уровень В

Варианты заданий

- В1. Найти значения и порядковые номера элементов, являющихся наименьшим среди четных чисел и наибольшим среди нечетных чисел.
- В2. Найти порядковый номер наибольшего по значению числа, являющегося симметричным в десятичном представлении.
- В3. Вычислить количество всех четных цифр во всех нечетных элементах последовательности.
- В4. Найти значение наибольшего элемента и его порядковый номер среди всех элементов, имеющих в своем составе цифру 3.
- В5. Найти количество чисел, которые не превосходят числа A и являются полными квадратами некоторого натурального числа.
- В6. Найти количество таких элементов, которые делятся на одну из своих цифр.
- В7. Найти количество таких элементов, сумма цифр в которых есть простое число.
- В8. Найти количество таких элементов, в которых чередуются четные и нечетные цифры.
- В9. Найти количество таких элементов, которые состоят из двух равных частей и имеют в своем составе цифру 3.
- В10. Найти в последовательности минимальное число, имеющее наибольшее количество цифр.
- В11. Найти такие элементы (а также их сумму), цифры в которых не превосходят заданного числа M ($0 < M < 9$).

В12. Найти такие элементы (а также их сумму), которые состоят только из четных цифр.

В13. Найти в последовательности число, имеющее в своем составе наибольшее количество четных цифр.

В14. Найти порядковый номер наибольшего по значению числа, являющегося симметричным в двоичном представлении.

В15. Найти сумму элементов последовательности, которые являются полными кубами некоторого натурального числа.

В16. Найти сумму таких элементов последовательности, которые делятся на сумму своих цифр.

В17. Вычислить общую сумму всех нечетных цифр во всех четных элементах.

Уровень С

В последовательности целых чисел найти максимальное количество чисел, идущих подряд, которые обладают свойством Q , и максимальное количество чисел, идущих подряд, которые не обладают свойством Q . Свойство Q задается в варианте. Программа должна содержать логическую функцию, проверяющую, обладает ли заданное число свойством Q .

Варианты заданий

С1. Q : число является простым.

С2. Q : число является симметричным в двоичном представлении.

С3. Q : число делится на одну из своих цифр.

С4. Q : число является полным квадратом некоторого натурального числа.

С5. Q : в числе чередуются четные и нечетные цифры.

С6. Q : число является симметричным в десятичном представлении.

С7. Q : число состоит из двух равных частей и имеет в своем составе цифру 1.

С8. Q : число состоит из попарно различных цифр.

С9. Q : цифры в числе образуют неубывающую последовательность.

С10. Q : сумма цифр числа является простым числом.

Лабораторная работа № 5. Определение характеристик одномерных массивов

Уровень А

Требуется определить массив целых чисел (например, размера 30), заполнить его случайными числами (в диапазоне от А до В, где А и В задаются в директивах #define) или ввести его элементы с клавиатуры и определить его характеристики в соответствии с вариантом.

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводимыми с клавиатуры;
- вывод массива на экран.

Варианты заданий

А1. В массиве целых чисел найти сумму элементов, расположенных между первым четным элементом и последним нечетным элементом массива.

А2. Найти количество и сумму тех элементов массива, которые делятся на 5 и не делятся на 7. Вывести все содержащиеся в данном массиве четные числа в порядке убывания их индексов, а также их количество.

А3. Найти количество тех элементов массива, которые больше своего левого соседа и меньше своего правого соседа. Вывести на экран номера минимальных элементов.

А4. Найти количество тех элементов массива, которые больше своего левого соседа. Найти количество и сумму положительных элементов массива, которые делятся на 3.

А5. Найти максимальный и минимальный элементы и номер элемента, наиболее близкого к среднему арифметическому минимального и максимального элементов.

А6. Вычислить сумму только симметричных элементов массива.

А7. Вычислить среднее арифметическое чисел, содержащих только четные цифры.

A8. Найти количество элементов, больших всех предыдущих и меньших всех последующих.

A9. В массиве целых чисел найти количество минимальных элементов.

A10. Вычислить сумму элементов массива, являющихся простыми числами и расположенных в интервале (L, R).

A11. Найти количество элементов, больших всех предыдущих элементов.

A12. В массиве целых чисел найти количество максимальных элементов.

A13. Найти сумму всех двузначных нечетных элементов целочисленного массива.

A14. Найти количество таких элементов целочисленного массива, которые имеют в младшем разряде цифру 7.

A15. Найти сумму элементов массива, меньших всех последующих элементов.

A16. Вычислить сумму элементов массива, содержащих хотя бы одну нечетную цифру.

A17. В массиве целых чисел найти количество простых чисел.

Пример программы уровня А. Найти количество всех положительных элементов целочисленного массива, сумма цифр которых больше 10.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 10
#define A -50
#define B 50
```

```
/* инициализация элементов массива а случайными числами в диапазоне от А до В */
```

```
void Init(int *a, int n)
{
    int i;
    for(i = 0; i < n; i++)
```

```

        a[i] = A + rand() % (B - A + 1);
    }

/* ВЫВОД МАССИВА НА ЭКРАН */
void Print(int *a, int n)
{
    int i;
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);
    putchar('\n');
}

/* СУММА ЦИФР ЧИСЛА a */
int Sum(int a)
{
    int sum = 0;
    while (a)
    {
        sum += a % 10;
        a /= 10;
    }
    return sum;
}

int main( )
{
    int a[N], i, count;
    srand(time(NULL));
    Init(a, N);
    Print(a, N);
    for(i = count = 0; i < N; i++)
        if (a[i] > 0 && Sum(a[i]) > 10)
            count++;
    printf("count = %d\n", count);
    return 0;
}

```

Уровень В

Требуется определить массив целых чисел (например, размера 30), заполнить его случайными числами (в диапазоне от А до В, где А и В задаются в директивах `#define`) или ввести его элементы с клавиатуры и определить его характеристики в соответствии с вариантом.

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводными с клавиатуры;
- вывод массива на экран.

Варианты заданий

В1. Найти порядковые номера и сумму двух попарно различных наибольших элементов.

В2. Найти наибольшее количество одинаковых элементов.

В3. Вычислить максимальное число подряд идущих положительных чисел, не превышающих числа М и имеющих в своем составе цифру 5.

В4. Найти количество попарно различных элементов массива, меньших числа М.

В5. Определить порядковые номера трех наименьших элементов и их сумму.

В6. Среди элементов массива найти индексы всех таких, которые могут быть получены суммированием двух элементов этого же массива.

В7. Напечатать все те элементы массива, сумма цифр в которых находится в интервале от L до R.

В8. Определить, имеется ли в массиве элемент, равный сумме наибольшего четного элемента и наименьшего нечетного элемента.

В9. Подсчитать количество элементов в массиве, являющихся точным квадратом другого элемента в этом массиве.

В10. Среди всех симметричных чисел найти наиболее длинное (по количеству цифр).

В11. Напечатать все те элементы массива, цифры которых образуют геометрическую прогрессию.

В12. Напечатать индексы элементов массива и сами элементы, сумма цифр которых кратна индексу этого элемента.

В13. Напечатать такие элементы массива и их количество, сумма цифр которых есть простое число.

В14. Определить, есть ли в числовом массиве элемент, взаимно простой с каждым из остальных элементов массива, и напечатать его.

В15. Среди элементов массива найти такие, которые могут быть получены с помощью произведения двух элементов этого же массива.

В16. Подсчитать сумму элементов в массиве, являющихся точным кубом другого элемента в этом массиве.

В17. Напечатать сумму и количество таких элементов массива, сумма цифр которых является составным числом.

Уровень С

Требуется определить массив целых чисел a (например, размера 30), заполнить его случайными числами (в диапазоне от A до B , где A и B задаются в директивах `#define`) или ввести его элементы с клавиатуры. Определить количество попарно различных элементов массива a , обладающих свойством Q (рассмотреть три случая: произвольный массив, упорядоченный массив, массив с достаточно узким диапазоном значений элементов). Свойства Q задаются в вариантах.

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводимыми с клавиатуры;
- вывод массива на экран;
- количество попарно различных элементов со свойством Q для произвольного массива;
- количество попарно различных элементов со свойством Q для упорядоченного массива;
- количество попарно различных элементов со свойством Q для массива с достаточно узким диапазоном значений элементов.

Варианты заданий

- C1. Q: число является четным.
- C2. Q: число является положительным.
- C3. Q: число содержит цифру 1.
- C4. Q: число не содержит цифру 0.
- C5. Q: число является нечетным.
- C6. Q: число является простым.
- C7. Q: число является двузначным.
- C8. Q: число является составным.
- C9. Q: число является нечетным.
- C10. Q: число содержит цифру 5.

Лабораторная работа № 6. Модификация одномерных массивов

Уровень А

Требуется определить массив целых чисел (например, размера 30), заполнить его случайными числами (в диапазоне от А до В, где А и В задаются в директивах #define) или ввести его элементы с клавиатуры и преобразовать массив в соответствии с вариантом задания.

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводимыми с клавиатуры;
- вывод массива на экран;
- вычисление минимального/максимального элемента массива.

Варианты заданий

А1. Заменить все положительные элементы массива на наименьший элемент массива.

А2. Заменить все элементы массива, находящиеся в интервале (L, R), на наибольший элемент массива.

А3. Заменить все отрицательные элементы массива на сумму минимального и максимального элементов.

А4. Заменить положительные четные элементы массива на наибольший элемент массива.

А5. Циклически сдвинуть массив на одну позицию влево и найти минимальный элемент.

А6. Заменить все элементы массива, которые делятся на 5 и не делятся на 7, на минимальный элемент.

А7. Заменить все нечетные значения целочисленного массива на модуль минимального элемента.

А8. Циклически сдвинуть массив на одну позицию вправо и найти максимальный элемент.

А9. Поменять порядок элементов массива на противоположный и найти сумму минимального и максимального элементов.

A10. Заменить все элементы массива, меньшие числа M , на наибольший элемент массива.

A11. Заменить отрицательные нечетные элементы массива на наименьший элемент массива.

A12. Заменить все элементы массива, которые делятся на 7, на максимальный элемент.

A13. Заменить все двузначные элементы массива на максимальный элемент.

A14. Заменить все нечетные двузначные элементы массива на наибольший элемент.

A15. Циклически сдвинуть массив на две позиции влево и найти минимальный элемент.

A16. Заменить все четные двузначные элементы массива на наименьший элемент.

A17. Циклически сдвинуть массив на две позиции вправо и найти максимальный элемент.

Пример программы уровня А. Заменить все положительные элементы массива, заканчивающиеся на цифру 3, на наименьший элемент массива.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 10
#define A -50
#define B 50
```

```
/* инициализация элементов массива а случайными числами в диапазоне от А до В */
```

```
void Init(int *a, int n)
{
    int i;
    for(i = 0; i < n; i++)
```

```

    a[i] = A + rand() % (B - A + 1);
}

/* ВЫВОД массива a на экран */
void Print(int *a, int n)
{
    int i;
    for(i = 0; i < n; i++)
        printf("%4d", a[i]);
    putchar('\n');
}

/* МИНИМАЛЬНЫЙ ЭЛЕМЕНТ массива a */
int Min(int *a, int n)
{
    int i, min = a[0];
    for(i = 1; i < n; i++)
        if (a[i] < min)
            min = a[i];
    return min;
}

int main( )
{
    int a[N], i, min;
    srand(time(NULL));
    Init(a, N);
    Print(a, N);
    min = Min(a, N);
    for(i = 0; i < N; i++)
        if (a[i] > 0 && a[i] % 10 == 3)
            a[i] = min;
    Print(a, N);
    return 0;
}

```

Уровень В

Требуется определить массив целых чисел (например, размера 30), заполнить его случайными числами (в диапазоне от А до В, где А и В задаются в директивах #define) или ввести его элементы с клавиатуры. Найти минимальный элемент массива из всех элементов, обладающих свойством Q. Все элементы массива, обладающие свойством Т, заменить на их обратные изображения (например, 123 заменить на 321). Отсортировать массив по возрастанию. После инициализации и каждого преобразования вывести массив на экран. Свойства Q и Т задаются в вариантах задания.

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводимыми с клавиатуры;
- вывод массива на экран;
- нахождение минимального элемента из всех элементов, обладающих свойством Q;
- сортировка элементов массива;
- целочисленная функция, которая возвращает число в перевернутом виде.

Варианты заданий

В1. Q: число является четным. Т: число содержит в своем составе цифры 2 и 3.

В2. Q: число является положительным. Т: число начинается с цифры 1.

В3. Q: число содержит цифру 1. Т: число является простым.

В4. Q: число не содержит цифру 0. Т: первая цифра числа совпадает с последней.

В5. Q: число является нечетным. Т: число является составным.

В6. Q: число является простым. Т: число содержит в своем составе цифру 5.

В7. Q: число является двузначным. Т: число не содержит в своем составе цифру 7.

В8. Q: число является симметричным. Т: число является трехзначным.

- В9. Q: сумма цифр числа равна 10. T: число является четным.
- В10. Q: число не содержит четных цифр. T: число является простым.
- В11. Q: число является составным. T: число начинается с 2.
- В12. Q: число начинается с 3. T: число является составным.
- В13. Q: число является степенью двойки. T: первая цифра числа является нечетной.
- В14. Q: число содержит хотя бы одну четную цифру. T: число содержит в младшем разряде цифру 3.
- В15. Q: число содержит цифру 3. T: число является составным.
- В16. Q: число является трехзначным. T: число содержит в своем составе цифру 2.
- В17. Q: число является степенью тройки. T: число не содержит в своем составе цифру 5.

Уровень С

Требуется определить массив целых чисел (например, размера 30), заполнить его случайными числами (в диапазоне от А до В, где А и В задаются в директивах `#define`) или ввести его элементы с клавиатуры. Упорядочить по возрастанию только те элементы целочисленного массива, которые обладают свойством Q, оставив остальные элементы на своих местах. Удалить из массива все элементы, обладающие свойством T. После инициализации и каждого преобразования выводить массив на экран. Свойства Q и T задаются в вариантах задания.

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводимыми с клавиатуры;
- вывод массива на экран;
- логическая функция, которая проверяет, обладает ли целое число свойством T;
- сортировка элементов массива, обладающих свойством Q;
- удаление элементов массива, обладающих свойством T.

Варианты заданий

- С1. Q: число является четным. Т: число состоит из попарно различных цифр.
- С2. Q: число является положительным. Т: только одна цифра в числе встречается ровно два раза, остальные – не более одного раза.
- С3. Q: число содержит цифру 1. Т: цифры в числе упорядочены по возрастанию.
- С4. Q: число не содержит цифру 0. Т: число является симметричными в двоичном представлении.
- С5. Q: число является нечетным. Т: число является простым и не содержит в своем составе цифру 5.
- С6. Q: число является простым. Т: в числе хотя бы одна цифра встречается более одного раза.
- С7. Q: число является двузначным. Т: число состоит только из простых цифр.
- С8. Q: число начинается на 1. Т: каждая цифра, входящая в число, встречается ровно два раза.
- С9. Q: сумма цифр числа является четным числом. Т: число является произведением двух различных простых чисел.
- С10. Q: число является квадратом некоторого натурального числа. Т: число является степенью некоторого простого числа.

Лабораторная работа № 7. Обработка текстовых строк

Требуется выделить в строке-предложении s все слова, разделенные символами-разделителями «`_.,:;\n\t!?`», и обработать выделенные слова в соответствии с вариантом задания.

Определения

Регулярное слово – слово, состоящее только из больших латинских букв.

Палиндром – это слово, которое одинаково читается слева направо и справа налево.

Алфавитный порядок задается таблицей ASCII.

Уровень А

Выделить в строке-предложении s все слова, разделенные символами-разделителями «`_.,:;\n\t!?`». Обработать выделенные слова в соответствии с вариантом.

Варианты заданий

- A1. Напечатать все слова, начинающиеся на большую и заканчивающиеся на маленькую букву.
- A2. Напечатать все слова, содержащие хотя бы одну цифру.
- A3. Напечатать все слова, содержащие хотя бы одну маленькую латинскую букву.
- A4. Напечатать все слова, содержащие хотя бы одну большую латинскую букву.
- A5. Напечатать все слова, состоящие только из маленьких латинских букв.
- A6. Напечатать все слова, состоящие только из больших латинских букв.
- A7. Напечатать все слова, состоящие только из цифр.
- A8. Напечатать все слова, состоящие не менее чем из четырех букв.
- A9. Напечатать все пятибуквенные слова.
- A10. Напечатать все слова, содержащие буквы и цифры.
- A11. Напечатать все слова, не содержащие ни одной цифры.

A12. Напечатать все слова, не содержащие ни одной большой латинской буквы.

A13. Напечатать все слова, содержащие хотя бы две цифры.

A14. Напечатать все слова, содержащие две рядом стоящие буквы.

A15. Напечатать все слова, содержащие две рядом стоящие цифры.

A16. Напечатать все слова, состоящие не более чем из пяти букв.

A17. Напечатать все слова, состоящие только из латинских букв.

Пример программы уровня А. Напечатать все слова, содержащие ровно две маленькие латинские буквы.

```
#include <stdio.h>
#include <ctype.h>
#define DELIMITERS " .,:;?!\\n\\t"
#define N 1024

/* количество маленьких латинских букв в строке (слове) s */
int Count(char *s)
{
    int i, count;
    for(i = count = 0; s[i] != '\\0'; i++)
        if (islower(s[i]))
            count++;
    return count;
}

int main( )
{
    char s[N];
    char *word;
    int i, j, flag[256] = {0};
    fgets(s, N, stdin);
    for (i = 0; DELIMITERS[i]; i++)
        flag[DELIMITERS[i]] = 1;
    for (i = 0; s[i] && flag[s[i]]; i++)
        ;
}
```

```

while (s[i])
{
    word = s + i;
    while (s[i] && !flag[s[i]])
        i++;
    j = i;
    while (s[i] && flag[s[i]])
        i++;
    s[j] = '\0';
    if (Count(word) == 2)
        puts(word);
}
return 0;
}

```

Уровень В

Выделить в строке-предложении *s* все слова, разделенные символами-разделителями «`_,,;\n\t!?`». Обработать выделенные слова в соответствии с вариантом задания. *Крайне не рекомендуется* использование функции `strtok()` из библиотеки `<string.h>`.

Варианты заданий

- В1. Подсчитать количество слов, начинающихся на большую букву и содержащих хотя бы один арифметический знак. Напечатать все слова, содержащие две рядом стоящие одинаковые буквы.
- В2. Подсчитать количество регулярных слов. Напечатать в перевернутом виде все слова, которые содержат два экземпляра заданного символа.
- В3. Напечатать все слова, начинающиеся с большой буквы. Напечатать самое длинное регулярное слово, которое состоит из одинаковых символов.
- В4. Напечатать слово, содержащее наибольшее количество цифр. Напечатать количество слов, содержащих хотя бы два арифметических знака.
- В5. Напечатать все регулярные слова. Напечатать в перевернутом виде самое длинное слово, состоящее только из цифр и латинских букв.

В6. Найти количество слов, содержащих более одной цифры, и, исключив все арифметические знаки из этих слов, напечатать их. Напечатать в порядке возрастания все числа, встретившиеся в словах.

В7. Определить количество слов, которые содержат заданное подслово и хотя бы одну цифру, и напечатать их. Напечатать в порядке убывания все числа, встретившиеся в словах.

В8. Определить количество слов, содержащих и буквы, и цифры, и арифметические знаки. Напечатать их. Напечатать все симметричные слова, содержащие наибольшее количество цифр.

В9. Подсчитать количество регулярных слов, содержащих хотя бы две одинаковые буквы. Напечатать все слова, имеющие одну цифру, удалив из таких слов все арифметические знаки.

В10. Найти самое длинное регулярное слово и удалить из него все гласные буквы. Найти все слова, в которых имеются либо только цифры, либо только латинские буквы.

В11. Подсчитать количество слов, состоящих из одинаковых букв или одинаковых цифр. Напечатать в перевернутом виде слова, имеющие хотя бы один арифметический знак.

В12. Напечатать все слова, которые начинаются с большой буквы и заканчиваются заданным двухбуквенным подсловом. Определить количество слов, содержащих согласные латинские буквы, и напечатать порядковые номера этих слов.

В13. Напечатать все слова, имеющие в своем составе согласные латинские буквы. Определить количество слов, которые не имеют в своем составе ни одной цифры, и напечатать эти слова.

В14. Напечатать все симметричные слова, предварительно удалив из них цифры. Напечатать все слова, состоящие только из согласных латинских букв.

В15. Найти все слова, содержащие числа в диапазоне от 10 до 100, и подсчитать их сумму. Напечатать слова, не имеющие цифр, предварительно удалив арифметические знаки.

V16. Подсчитать количество слов, начинающихся с большой буквы и оканчивающихся цифрой. Напечатать слова, содержащие заданное подслово и хотя бы один арифметический знак.

V17. Подсчитать количество слов, содержащих хотя бы одну согласную латинскую букву и хотя бы одну цифру. Напечатать все слова, состоящие только из четных цифр, и подсчитать сумму этих цифр.

V18. Напечатать все слова, которые содержат хотя бы один арифметический знак и заканчиваются на цифру. Определить количество слов, содержащих все маленькие латинские гласные буквы.

V19. Найти количество симметричных регулярных слов и напечатать их. Напечатать в перевернутом виде все слова, содержащие согласные латинские буквы.

V20. Найти и напечатать все слова, содержащие наибольшее количество букв, если только буквы расположены в алфавитном порядке. Подсчитать количество симметричных слов, имеющих более двух арифметических знаков.

Уровень С

Выделить в строке-предложении s все слова, разделенные символами-разделителями «`_.,:;\n\t!?`». Найти количество всех слов в строке-предложении, которые обладают свойством Q. Вывести все слова исходной строки-предложения в лексикографическом порядке. Записать в строку t любое из слов максимальной длины строки s . Удалить из строки t все символы, обладающие свойством T. *Запрещается* использование функции `strtok()` из библиотеки `<string.h>`.

Программа должна содержать следующие функции:

- сортировка слов (указателей);
- удаление символов строки, обладающих свойством T.

Варианты заданий

S1. Q: в слове нет повторяющихся символов. T: символ является маленькой гласной латинской буквой.

S2. Q: в слове все символы встречаются более одного раза. T: символ является большой гласной латинской буквой.

C3. Q: в слове хотя бы один символ встречается более одного раза. T: символ является маленькой согласной латинской буквой.

C4. Q: в слове только один символ встречается более одного раза. T: символ является большой согласной латинской буквой.

C5. Q: в слове латинские буквы чередуются с символами (например, «a2b5s»). T: символ является цифрой.

C6. Q: в слове гласные латинские буквы чередуются с согласными. T: символ является арифметическим знаком.

C7. Q: в слове строчные латинские буквы чередуются с прописными. T: символ является простой цифрой.

C8. Q: в слове буквы располагаются по возрастанию и не повторяются. T: символ является маленькой согласной латинской буквой.

C9. Q: в слове гласные латинские буквы чередуются с цифрами. T: символ является маленькой гласной латинской буквой.

C10. Q: в слове согласные латинские буквы чередуются с четными цифрами. T: символ является большой согласной латинской буквой.

Лабораторная работа № 8. Обработка и модификация двумерных массивов

Уровень А

Требуется определить двумерный массив a целых чисел (например, размера 5×7), заполнить его случайными числами или ввести его элементы с клавиатуры и в одномерный массив b записать характеристики двумерного массива a в соответствии с вариантом задания.

Программа должна содержать следующие функции:

- инициализация элементов двумерного массива случайными числами или вводимыми с клавиатуры;
- заполнение одномерного массива в соответствии с заданием;
- вывод одномерного массива на экран;
- вывод двумерного массива на экран.

Варианты заданий

A1. Найти суммы четных положительных элементов каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A2. Найти количество четных положительных элементов каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A3. Найти суммы нечетных положительных элементов каждого столбца матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A4. Найти количество нечетных положительных элементов каждого столбца матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A5. Найти суммы положительных элементов, делящихся на 3, каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A6. Найти количество положительных элементов, делящихся на 5, каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A7. Найти суммы положительных элементов, не делящихся на 3, каждого столбца матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A8. Найти количество положительных элементов, не делящихся на 5, каждого столбца матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A9. Найти минимальные элементы каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A10. Найти минимальные элементы каждого столбца матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A11. Найти количество двузначных элементов каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A12. Найти максимальные элементы каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A13. Найти максимальные элементы каждого столбца матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A14. Найти суммы двузначных элементов каждого столбца матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A15. Найти количество минимальных элементов каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A16. Найти количество максимальных элементов каждого столбца матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

A17. Найти количество простых чисел каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

Пример программы уровня А. Найти количество положительных элементов, заканчивающихся на 1, каждой строки матрицы $a(5, 7)$ и сохранить их в одномерном массиве b .

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#define M 5
#define N 7
#define A -50
#define B 50
```

```
/* инициализация элементов матрицы а случайными числами в диапазоне от А до В */
```

```
void InitA(int a[][N], int m, int n)
```

```

{
    int i, j;
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            a[i][j] = A + rand() % (B - A + 1);
}

```

```

/* ВЫВОД МАТРИЦЫ НА ЭКРАН */
void PrintA(int a[][N], int m, int n)

```

```

{
    int i, j;
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
            printf("%5d", a[i][j]);
        printf("\n");
    }
    printf("\n");
}

```

```

/* ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ ЭЛЕМЕНТОВ МАССИВА b */
void InitB(int a[][N], int m, int n, int *b)

```

```

{
    int i, j, count;
    for(i = 0; i < m; i++)
        for(j = count = 0; j < n; j++)
        {
            if (a[i][j] > 0 && a[i][j] % 10 == 1)
                count++;
            b[i] = count;
        }
}

```

```

/* ВЫВОД МАССИВА НА ЭКРАН */
void PrintB(int *a, int n)

```

```

{
    int i;

```

```

    for(i = 0; i < n; i++)
        printf("%5d ", a[i]);
    printf("\n");
}

int main( )
{
    int a[M][N], b[M];
    srand(time(NULL));
    InitA(a, M, N);
    PrintA(a, M, N);
    InitB(a, M, N, b);
    PrintB(b, M);
    return 0;
}

```

Уровень В

Требуется определить двумерный массив целых чисел (например, размера 5×7), заполнить его случайными числами (в диапазоне от А до В, где А и В задаются в директивах `#define`) или ввести его элементы с клавиатуры и преобразовать массив в соответствии с вариантом.

Программа должна содержать следующие функции:

- инициализация элементов двумерного массива случайными числами или вводимыми с клавиатуры;
- вывод двумерного массива на экран.

Варианты заданий

В1. Задан прямоугольный массив целых чисел. Определить, есть ли такой элемент, который входит во все строки данного массива, но не во все столбцы, и напечатать его.

В2. Символьный квадратный массив заполнен случайным набором символов. Определить количество квадратиков, образованных совокупностью одинаковых символов, и их суммарную площадь. Квадратик – это образование из не менее чем $4=2 \times 2$ одинаковых рядом стоящих символов. Разные квадратики могут быть заполнены разными символами.

В3. В квадратный числовой массив записаны целые числа в диапазоне от А

до В. Определить, имеются ли такие квадратные подмассивы, сумма элементов которых равна Р. Если таковые имеются, то вывести эти подмассивы на экран.

В4. Задан прямоугольный массив целых чисел. Определить, все ли строки состоят только из разных чисел, и напечатать строки, имеющие в своем составе одинаковые числа.

В5. Квадратный числовой массив моделирует расположение непересекающихся прямоугольников в квадрате, причем каждый прямоугольник заполнен своим числом, отличным от нуля, а промежутки между ними заполнены нулями. Определить количество прямоугольников и их общую площадь.

В6. В квадратный числовой массив записаны целые числа в диапазоне от А до В. Отсортировать по убыванию элементы массива по строкам, а затем строки по убыванию первых элементов в строке. Результирующий массив вывести на экран.

В7. В логическом квадратном массиве порядка 8, моделирующем шахматную доску, значениям 1 соответствуют клетки, на которых расположены фигуры коня, 0 соответствует пустой клетке. Определить, имеются ли на доске фигуры, угрожающие другим, и напечатать их координаты.

В8. В числовом прямоугольном массиве $M \times N$ каждый элемент равен числу вида $100 \cdot i + j$, где $0 \leq i \leq M-1$, $0 \leq j \leq N-1$, или нулю. Это число показывает, на какую следующую клетку (i, j) массива следует перейти с данной клетки. Нуль означает конец пути. Определить, можно ли таким образом посетить все клетки массива, начав с элемента $(0, 0)$. Напечатать координаты начальной и конечной клеток, а также количество посещений.

В9. В прямоугольной матрице найти количество строк, элементы которых упорядочены по возрастанию.

В10. В частично заполненном значением 1 логическом квадратном массиве имеются свободные участки, моделируемые значением 0. Определить, можно ли разместить на них квадрат из значений 1 размером М. Если возможно, то напечатать массив со вставленным квадратом на экране.

В11. В логическом квадратном массиве порядка 8, моделирующем шахматную доску, значениям 1 соответствуют клетки, на которых расположены фигуры ферзя, 0 соответствует пустой клетке. Определить, сколько

имеется на доске фигур, угрожающих другим, и удалить с доски те из них, которые угрожают двум или более фигурам.

В12. В логическом квадратном массиве порядка 8, моделирующем шахматную доску, значениям 1 соответствуют клетки, на которых расположены фигуры слона, 0 соответствует пустой клетке. Определить, сколько имеется на доске фигур, угрожающих другим, и удалять их с доски до тех пор, пока слоны не будут угрожать друг другу. Напечатать количество оставшихся фигур. Результирующий массив вывести на экран.

В13. В прямоугольной матрице найти количество столбцов, элементы которых упорядочены по убыванию.

В14. Упорядочить строки прямоугольной матрицы так, чтобы их первые элементы образовывали возрастающую последовательность.

В15. В прямоугольной матрице найти количество столбцов, которые состоят из попарно различных элементов.

В16. В прямоугольной матрице найти номер первого из ее столбцов, содержащих максимальное количество одинаковых элементов.

В17. В прямоугольной целочисленной матрице найти номер последней из ее строк, содержащих максимальное количество простых чисел.

Уровень С

Требуется определить двумерный массив целых чисел (например, размера 5×7), заполнить его случайными числами (в диапазоне от А до В, где А и В задаются в директивах `#define`) или ввести его элементы с клавиатуры и преобразовать массив в соответствии с вариантом задания.

Программа должна содержать следующие функции:

- инициализация элементов двумерного массива случайными числами или вводимыми с клавиатуры;
- вывод двумерного массива на экран.

Варианты заданий

С1. В квадратный числовой массив записаны целые числа в диапазоне от А до В. Отсортировать по убыванию элементы массива, расположенные по часовой стрелке по спирали, начиная с верхнего левого элемента. Результирующий массив вывести на экран.

C2. Символьный квадратный массив заполнен случайным набором символов. Найти цепочки, расположенные по вертикали и горизонтали и состоящие из цифр, преобразовать их в числа и напечатать на экране эти числа и их сумму.

C3. Квадратный массив, элементами которого являются английские буквы и символ пробела, заполнен кроссвордом. Напечатать все слова в кроссворде с чередующимися гласными и согласными буквами и их количество.

C4. Квадратный массив, элементами которого являются английские буквы и символ пробела, заполнен кроссвордом. Напечатать общее количество слов в кроссворде и все буквы, которые находятся на пересечении слов кроссворда.

C5. В числовом квадратном массиве построить «спираль Улама» следующим образом: начиная с центрального элемента по спирали против часовой стрелки массив заполняется натуральными числами от 1 до N^2 . Определить количество простых чисел в массиве.

C6. Две строки матрицы назовем эквивалентными, если совпадают множества элементов, встречающихся в этих строках, например, строки «1, 2, 1» и «2, 1, 2» эквивалентны. Найти количество строк матрицы, эквивалентных первой строке.

C7. Найти минимальный среди элементов тех строк, которые упорядочены либо по возрастанию, либо по убыванию. Если упорядоченные строки в матрице отсутствуют, то вывести сообщение об этом.

C8. Найти максимальный среди элементов тех столбцов, которые упорядочены либо по возрастанию, либо по убыванию. Если упорядоченные столбцы в матрице отсутствуют, то вывести сообщение об этом.

C9. Упорядочить строки матрицы так, чтобы их минимальные элементы образовывали убывающую последовательность.

C10. Два столбца матрицы назовем эквивалентными, если совпадают множества элементов, встречающихся в этих столбцах, например, столбцы «1,2,1» и «2,1,2» эквивалентны. Найти количество столбцов матрицы, эквивалентных первому столбцу.

Лабораторная работа № 9. Рекурсия

Уровень А

Требуется определить массив целых чисел (например, размера 30), заполнить его случайными числами или ввести его элементы с клавиатуры и определить его характеристики с помощью рекурсивной функции в соответствии с вариантом задания. Проверку правильности результата провести с помощью сравнения результатов итеративной и рекурсивной функций.

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводимыми с клавиатуры;
- вывод массива на экран;
- итеративная функция, вычисляющая характеристики массива в соответствии с вариантом задания;
- рекурсивная функция, вычисляющая характеристики массива в соответствии с вариантом задания.

Варианты заданий

A1. В массиве целых чисел с помощью рекурсивной функции найти количество четных элементов.

A2. В массиве целых чисел с помощью рекурсивной функции найти количество элементов массива, которые делятся на 5 и не делятся на 7.

A3. В массиве целых чисел с помощью рекурсивной функции найти количество простых чисел.

A4. В массиве целых чисел с помощью рекурсивной функции найти сумму нечетных элементов.

A5. Найти максимальный элемент целочисленного массива с помощью рекурсии.

A6. Вычислить сумму только симметричных элементов массива с помощью рекурсии.

A7. В массиве целых чисел с помощью рекурсивной функции найти сумму таких элементов, которые состоят только из четных цифр.

A8. В массиве целых чисел с помощью рекурсивной функции найти количество элементов, больших заданного числа A.

A9. В массиве целых чисел с помощью рекурсии найти количество минимальных элементов.

A10. В массиве целых чисел с помощью рекурсии найти сумму элементов массива, расположенных в интервале (A, B).

A11. Найти количество двузначных элементов массива с помощью рекурсии.

A12. Найти минимальный элемент целочисленного массива с помощью рекурсии.

A13. В массиве целых чисел с помощью рекурсии найти количество максимальных элементов.

A14. Вычислить сумму только двузначных элементов массива с помощью рекурсии.

A15. В массиве целых чисел с помощью рекурсивной функции найти сумму элементов, заканчивающихся на 7.

A16. В массиве целых чисел с помощью рекурсивной функции найти сумму всех элементов массива, которые одновременно делятся на 3 и на 5.

A17. В массиве целых чисел с помощью рекурсивной функции найти количество элементов, начинающихся на 3.

Пример программы уровня А. В массиве целых чисел с помощью рекурсии найти количество таких элементов, первая цифра которых совпадает с последней.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#define N 10
#define A -50
#define B 50
```

```
/* инициализация элементов массива а случайными числами в диапазоне от А до В */
```

```
void Init(int *a, int n)
{
    int i;
    for(i = 0; i < n; i++)
        a[i] = A + rand() % (B - A + 1);
}
```

```
/* вывод массива на экран */
```

```
void Print(int *a, int n)
{
    int i;
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);
    putchar('\n');
}
```

```
/* проверка, совпадает ли первая цифра числа а с последней */
```

```
int Check(int a)
{
    int b = abs(a), c = b % 10;
    while (b > 9)
        b /= 10;
    return (b == c);
}
```

```
/* рекурсивная функция, вычисляющая количество таких элементов массива, первая цифра которых совпадает с последней */
```

```
int CountRec(int *a, int *end)
{
    return a < end ? Check(*a) + CountRec(a + 1, end) : 0;
}
```

```

/* итеративная функция, вычисляющая количество таких элементов массива,
первая цифра которых совпадает с последней */
int CountIt(int *a, int n)
{
    int i, count;
    for(i = count = 0; i < n; i++)
        count += Check(a[i]);
    return count;
}

int main( )
{
    int a[N], i, count;
    srand(time(NULL));
    Init(a, N);
    Print(a, N);
    printf("count1 = %d\n", CountRec(a, a + N));
    printf("count2 = %d\n", CountIt(a, N));
    return 0;
}

```

Уровень В

Метод «разделяй и властвуй»

Требуется определить одномерный целочисленный массив a из n элементов (например, $n=30$), заполнить его случайными числами (в диапазоне от A до B , где A и B задаются в директивах `#define`) или ввести его элементы с клавиатуры и определить следующие его характеристики с помощью метода «разделяй и властвуй»:

- количество элементов массива, обладающих свойством Q ;
- сумму элементов массива, обладающих свойством T ;
- обладает ли хотя бы один элемент массива a свойством Q ;
- все ли элементы массива a обладают свойством T .

Свойства Q и T задаются в индивидуальных вариантах. Проверку правильности результата провести с помощью сравнения результатов итеративной и рекурсивной функций.

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводимыми с клавиатуры;
- вывод массива на экран;
- итеративная функция, вычисляющая количество элементов массива, обладающих свойством Q;
- рекурсивная функция, вычисляющая количество элементов массива, обладающих свойством Q;
- итеративная функция, вычисляющая сумму элементов массива, обладающих свойством T;
- рекурсивная функция, вычисляющая сумму элементов массива, обладающих свойством T;
- итеративная логическая функция, проверяющая, обладает ли хотя бы один элемент массива свойством Q;
- рекурсивная логическая функция, проверяющая, обладает ли хотя бы один элемент массива свойством Q;
- итеративная логическая функция, проверяющая, все ли элементы массива a обладают свойством T;
- рекурсивная логическая функция, проверяющая, все ли элементы массива a обладают свойством T.

Варианты заданий

В1. Q: число является четным. T: число содержит в своем составе цифры 2 и 3.

В2. Q: число является положительным. T: число начинается с цифры 1.

В3. Q: число содержит цифру 1. T: число является простым.

В4. Q: число не содержит цифру 0. T: первая цифра числа совпадает с последней.

В5. Q: число является нечетным. T: число является составным.

В6. Q: число является простым. T: число содержит в своем составе цифру 5.

В7. Q: число является двузначным. T: число не содержит в своем составе цифру 7.

- В8. Q: число является симметричным. Т: число является трехзначным.
- В9. Q: сумма цифр числа равна 10. Т: число является четным.
- В10. Q: число не содержит четных цифр. Т: число является простым.
- В11. Q: число является составным. Т: число начинается с 2.
- В12. Q: число начинается с 3. Т: число является составным.
- В13. Q: число является степенью двойки. Т: первая цифра числа является нечетной.
- В14. Q: число содержит хотя бы одну четную цифру. Т: число содержит в младшем разряде цифру 3.
- В15. Q: число содержит цифру 3. Т: число является составным.
- В16. Q: число является трехзначным. Т: число содержит в своем составе цифру 2.
- В17. Q: число является степенью тройки. Т: число не содержит в своем составе цифру 5.

Уровень С

Метод «разделяй и властвуй»

Требуется определить одномерный целочисленный массив a из n элементов (например, $n=30$), заполнить его случайными числами (в диапазоне от A до B , где A и B задаются в директивах `#define`) или ввести его элементы с клавиатуры и определить следующие его характеристики с помощью метода «разделяй и властвуй»:

- количество элементов массива, обладающих свойством Q;
- сумму элементов массива, обладающих свойством T;
- обладает ли хотя бы один элемент массива a свойством Q;
- все ли элементы массива a обладают свойством T.

Свойства Q и T задаются в индивидуальных вариантах. Проверку правильности результата провести с помощью сравнения результатов итеративной и рекурсивной функций. Также требуется упорядочить массив по возрастанию с помощью рекурсивной функции (алгоритм сортировки задается в варианте).

Программа должна содержать следующие функции:

- инициализация элементов массива случайными числами или вводимыми с клавиатуры;
- вывод массива на экран;
- итеративная функция, вычисляющая количество элементов массива, обладающих свойством Q;
- рекурсивная функция, вычисляющая количество элементов массива, обладающих свойством Q;
- итеративная функция, вычисляющая сумму элементов массива, обладающих свойством T;
- рекурсивная функция, вычисляющая сумму элементов массива, обладающих свойством T;
- итеративная логическая функция, проверяющая, обладает ли хотя бы один элемент массива свойством Q;
- рекурсивная логическая функция, проверяющая, обладает ли хотя бы один элемент массива свойством Q;
- итеративная логическая функция, проверяющая, все ли элементы массива a обладают свойством T;
- рекурсивная логическая функция, проверяющая, все ли элементы массива a обладают свойством T;
- рекурсивная сортировка элементов массива.

Варианты заданий

C1. Q: число является четным. T: число состоит из попарно различных цифр. Рекурсивно упорядочить массив с помощью метода прямого выбора.

C2. Q: число является положительным. T: только одна цифра в числе встречается ровно два раза, остальные – не более одного раза. Рекурсивно упорядочить массив с помощью метода вставок.

C3. Q: число содержит цифру 1. T: цифры в числе упорядочены по возрастанию. Рекурсивно упорядочить массив с помощью пузырьковой сортировки.

C4. Q: число не содержит цифру 0. T: число является симметричными в двоичном представлении. Рекурсивно упорядочить массив с помощью метода прямого выбора.

С5. Q: число является нечетным. T: число является простым и не содержит в своем составе цифру 5. Рекурсивно упорядочить массив с помощью метода вставок.

С6. Q: число является простым. T: в числе хотя бы одна цифра встречается более одного раза. Рекурсивно упорядочить массив с помощью пузырьковой сортировки.

С7. Q: число является двузначным. T: число состоит только из простых цифр. Рекурсивно упорядочить массив с помощью метода прямого выбора.

С8. Q: число начинается на 1. T: каждая цифра, входящая в число, встречается ровно два раза. Рекурсивно упорядочить массив с помощью метода вставок.

С9. Q: сумма цифр числа является четным числом. T: число является произведением двух различных простых чисел. Рекурсивно упорядочить массив с помощью пузырьковой сортировки.

С10. Q: число является квадратом некоторого натурального числа. T: число является степенью некоторого простого числа. Рекурсивно упорядочить массив с помощью метода вставок.

Лабораторная работа № 10. Текстовые файлы

Текстовый файл – это последовательность строк, причем каждая строка заканчивается символом «\n». Текстовый файл состоит из обычных печатных символов (в кодах ASCII), включая пробелы, символы новой строки, символы табуляции. Например, в текстовом файле целое число 123 представлено тремя символами-цифрами «1», «2» и «3», которые в файле расположены друг за другом, образуя слово «123».

Требуется составить программу, выполняющую преобразования над текстовыми файлами согласно указанному варианту.

Уровень А

Варианты заданий

- A1. Напечатать все слова из текстового файла, начинающиеся на большую и заканчивающиеся на маленькую букву.
- A2. Напечатать все слова из текстового файла, содержащие хотя бы одну цифру.
- A3. Напечатать все слова из текстового файла, содержащие хотя бы одну маленькую латинскую букву.
- A4. Напечатать все слова из текстового файла, содержащие хотя бы одну большую латинскую букву.
- A5. Напечатать все слова из текстового файла, состоящие только из маленьких латинских букв.
- A6. Напечатать все слова из текстового файла, состоящие только из больших латинских букв.
- A7. Напечатать все слова из текстового файла, состоящие только из цифр.
- A8. Напечатать все слова из текстового файла, состоящие не менее чем из четырех букв.
- A9. Напечатать все пятибуквенные слова из текстового файла.
- A10. Напечатать все слова из текстового файла, содержащие буквы и цифры.
- A11. Напечатать все слова из текстового файла, не содержащие ни одной цифры.

A12. Напечатать все слова из текстового файла, не содержащие ни одной большой латинской буквы.

A13. Напечатать все слова из текстового файла, содержащие ровно две цифры.

A14. Напечатать все слова из текстового файла, содержащие две рядом стоящие одинаковые буквы.

A15. Напечатать все слова из текстового файла, содержащие две рядом стоящие цифры.

A16. Напечатать все слова из текстового файла, состоящие не более чем из пяти букв.

A17. Напечатать все слова из текстового файла, состоящие только из латинских букв.

Пример программы уровня А. Напечатать все слова из текстового файла, разделенные символами-разделителями «_.,;:\n\t!?»», содержащие ровно две большие латинские буквы.

```
#include <stdio.h>
#include <ctype.h>
#define DELIMITERS " _.,;?!\\n\\t"
#define N 1024

/* количество больших латинских букв */
int Count(char *s)
{
    int i, count;
    for(i = count = 0; s[i] != '\0'; i++)
        if (isupper(s[i]))
            count++;
    return count;
}

/* вывод всех слов с требуемым условием */
int ReadWords(char *fname)
```

```

{
    FILE *f;
    char s[N], *word;
    int i, j, flag[256] = {0};
    if ((f = fopen(fname, "r")) == NULL)
        return 1;
    for (i = 0; DELIMITERS[i]; i++)
        flag[DELIMITERS[i]] = 1;
    while (fgets(s, N, f) != NULL)
    {
        for (i = 0; s[i] && flag[s[i]]; i++)
            ;
        while (s[i])
        {
            word = s + i;
            while (s[i] && !flag[s[i]])
                i++;
            j = i;
            while (s[i] && flag[s[i]])
                i++;
            s[j] = '\0';
            if (Count(word) == 2)
                puts(word);
        }
    }
    fclose(f);
    return 0;
}

int main( )
{
    ReadWords("c:\\a.txt");
    return 0;
}

```

Уровень В

Выделить в текстовом файле все слова, разделенные символами-разделителями «_.,;:\n\t!?»», и все слова, обладающие свойством Q, построчно записать в другой текстовый файл. Свойства Q задаются в индивидуальных вариантах.

Варианты заданий

- В1. Q: в слове нет арифметических знаков.
- В2. Q: в слове нет гласных латинских букв.
- В3. Q: слово состоит только из согласных латинских букв.
- В4. Q: в слове присутствуют и цифры, и арифметические знаки.
- В5. Q: в слове присутствуют и гласные, и согласные латинские буквы.
- В6. Q: в слове нет согласных латинских букв и цифр.
- В7. Q: в слове нет гласных латинских букв и арифметических знаков.
- В8. Q: в слове присутствуют гласные латинские буквы.
- В9. Q: в слове присутствуют согласные латинские буквы, но нет цифр.
- В10. Q: в слове присутствуют цифры, но нет арифметических знаков.
- В11. Q: в слове согласных латинских букв больше чем гласных.
- В12. Q: в слове количество цифр равно количеству арифметических знаков.
- В13. Q: в слове гласных латинских букв больше чем цифр.
- В14. Q: в слове количество гласных латинских букв равно количеству арифметических знаков.
- В15. Q: слово состоит только из гласных латинских букв и цифр.
- В16. Q: слово состоит только из согласных латинских букв и является регулярным.
- В17. Q: в слове нет согласных латинских букв, и оно является симметричным.

Уровень С

Выделить в текстовом файле все слова, разделенные символами-разделителями «_.,;\n\t!?»». Найти количество всех слов, которые обладают свойством Q. Все слова из исходного файла записать во второй файл построчно в алфавитном порядке в следующем виде:

<слово> – <количество повторений в исходном файле>.

Варианты заданий.

- C1. Q: в слове нет повторяющихся символов.
- C2. Q: в слове все символы встречаются более одного раза.
- C3. Q: в слове хотя бы один символ встречается более одного раза.
- C4. Q: в слове только один символ встречается более одного раза.
- C5. Q: в слове латинские буквы чередуются с цифрами (например, «a2b5s»).
- C6. Q: в слове гласные латинские буквы чередуются с согласными.
- C7. Q: в слове строчные латинские буквы чередуются с прописными.
- C8. Q: в слове буквы располагаются по возрастанию и не повторяются.
- C9. Q: в слове гласные латинские буквы чередуются с цифрами.
- C10. Q: в слове согласные латинские буквы чередуются с нечетными цифрами.

Лабораторная работа № 11. Двоичные файлы. Базы данных

Уровень А

Разработать программу, реализующую простейшую базу данных, со следующими функциями:

- CreateFile(char *fileName) – создание двоичного файла данных, элементами которого являются записи со структурой, указанной в индивидуальном варианте.
- ReadFile(char *fileName) – просмотр записей во входном файле последовательно от первой до последней.
- Count(char *fileName) – подсчет количества записей в исходном файле в соответствии с вариантом.

Варианты заданий

А1. АВТОМОБИЛИ

Входной файл содержит сведения об автомобилях:

```
struct Rec
{
    char mark[20]; // марка автомобиля
    long mas;      // масса автомобиля в тоннах
    int year;      // год выпуска
    double price; // цена
};
```

Найти количество самых новых автомобилей.

А2. КАТАЛОГ СПОРТИВНЫХ ТОВАРОВ

Входной файл содержит сведения о спортивных товарах:

```
struct Rec
{
    char name[20]; // наименование
    char type[20]; // вид спорта
    double price; // цена
};
```

Найти количество самых дорогих товаров.

А3. КАТАЛОГ ЧАЙНОЙ ПРОДУКЦИИ

Входной файл содержит сведения о чайной продукции:

```
struct Rec
{
    char name[20]; // производитель
    long mas;      // масса
    char type[20]; // тип чая (зеленый, черный,...)
    double price;  // цена
};
```

Найти количество самых дорогих сортов чая.

А4. КАТАЛОГ МОЛОЧНОЙ ПРОДУКЦИИ

Входной файл содержит сведения о молочной продукции:

```
struct Rec
{
    char name[20]; // производитель
    long mas;      // масса
    char type[20]; // тип продукции (молоко, творог...)
    double price;  // цена
};
```

Найти количество творожной продукции в каталоге.

А5. КАТАЛОГ МЕБЕЛЬНОЙ ПРОДУКЦИИ

Входной файл содержит сведения о мебельной продукции:

```
struct Rec
{
    char name[20]; // наименование
    long mas;      // масса
    char type[20]; // сорт древесины
    double price;  // цена
};
```

Найти количество самой тяжелой мебельной продукции.

А6. КАТАЛОГ ЧАСОВ

Входной файл содержит сведения о часах:

```

struct Rec
{
    char name[20]; // производитель (наименование)
    char type[20]; // тип часов (механические, кварцевые...)
    double price; // цена
};

```

Найти количество механических часов в каталоге.

А7. ВИДЕОТЕКА

Входной файл содержит сведения о фильмах:

```

struct Rec
{
    char name[50]; // название фильма
    char zhanr[20]; // жанр фильма
    long time; // продолжительность фильма
};

```

Найти количество самых продолжительных фильмов.

А8. АУДИОТЕКА

Входной файл содержит сведения о музыкальных произведениях:

```

struct Rec
{
    char name[50]; // исполнитель (название группы)
    char zhanr[20]; // направление жанра
    long time; // продолжительность в мин.
};

```

Найти количество самых коротких музыкальных произведений.

А9. МОТОЦИКЛЫ

Входной файл содержит сведения о мотоциклах:

```

struct Rec
{
    char mark[20]; // марка мотоцикла
    long mas; // масса
    int year; // год выпуска
};

```

```
double price; // цена
};
```

Найти количество самых тяжелых мотоциклов.

A10. КАТАЛОГ ЮВЕЛИРНЫХ УКРАШЕНИЙ

Входной файл содержит сведения о ювелирных украшениях:

```
struct Rec
{
    char name[30]; // наименование
    char type[30]; // материал (золото, серебро...)
    long mas;      // масса
    double price; // цена
};
```

Найти количество золотых украшений.

A11. КАТАЛОГ КАНЦЕЛЯРСКИХ ТОВАРОВ

Входной файл содержит сведения о канцелярских товарах:

```
struct Rec
{
    char name[30]; // наименование
    long mas;      // масса
    double price; // цена
};
```

Найти количество тетрадей в каталоге.

A12. БУТИК

Входной файл содержит сведения о товарах некоторого бутика:

```
struct Rec
{
    char name[30]; // наименование
    char country[30]; // страна-производитель
    long size;     // размер
    double price; // цена
};
```

Найти количество товаров из заданной страны.

A13. КОНДИТЕРСКИЕ ИЗДЕЛИЯ

Входной файл содержит сведения о кондитерских изделиях:

```
struct Rec
{
    char name[30];    // наименование
    long mas;        // вес
    double price;    // цена
};
```

Найти количество самых дешевых кондитерских изделий.

A14. БЫТОВАЯ ТЕХНИКА

Входной файл содержит сведения о товарах некоторого бутика:

```
struct Rec
{
    char name[30];    // наименование
    char country[30]; // страна-производитель
    long mas;        // вес
    double price;    // цена
};
```

Найти количество самых дорогих товаров.

A15. ТОЧКИ ПЛОСКОСТИ

Входной файл содержит координаты точек трехмерного пространства:

```
struct Rec
{
    double x;    // абсцисса
    double y;    // ордината
    double z;    // аппликата
};
```

Найти количество тех точек плоскости, которые содержатся внутри окружности с центром в начале координат и радиуса R.

A16. ПРАЗДНИЧНЫЕ ДНИ

Входной файл содержит сведения о праздничных днях:

```
struct Rec
```

```

{
    char name[30];    // название праздника
    int day;         // день
    char month[30];  // месяц
};

```

Найти количество праздничных дней, которые имеются в заданном месяце.

Уровень В

Разработать программу, основной блок которой в режиме «МЕНЮ» позволяет выбрать и запустить рабочие функции следующего назначения:

- CreateFileIn(char *fileName) – создание двоичного файла данных, элементами которого являются записи со структурой, указанной в индивидуальном варианте.
- AddFileIn(char *fileName) – добавление записей в конец входного файла.
- ReadFileIn(char *fileName) – просмотр записей во входном файле последовательно от первой до последней.
- CreateFileOut(char *fileName1, char *fileName2) – обработка данных из входного файла по заданию индивидуального варианта и вывод результатов в выходной файл.
- ReadFileOut(char *fileName) – просмотр записей в выходном файле последовательно от первой до последней.

Варианты заданий

В1. КНИЖНЫЙ МАГАЗИН

Входной файл содержит сведения о книгах:

```

struct Rec1
{
    int nomer;        // номер записи в файле
    char name[30];    // фамилия автора
    char nazv[30];    // название книги
    int year;         // год издания
    float price;      // цена
};

```

Требуется в выходной файл записать все те записи о книгах, которые являются самыми дорогими:

```
struct Rec2
{
    char name[30];    // фамилия автора
    char nazv[30];   // название книги
    int year;        // год издания
    float price;     // цена
};
```

В2. ХАРАКТЕРИСТИКИ ПК

Входной файл содержит сведения о персональных компьютерах:

```
struct Rec1
{
    int nomer;       // номер записи в файле
    char cpu[30];    // марка процессора
    int frecuency;   // тактовая частота
    int ddr;         // объем ОП в Мбайт
    int ssd;         // объем SSD в Гбайт
    float price;     // цена
};
```

Требуется в выходной файл записать сведения о всех тех ПК, которые имеют наибольшую тактовую частоту:

```
struct Rec2
{
    char cpu[30];    // марка процессора
    int frecuency;   // тактовая частота
    int ddr;         // объем ОП в Мбайт
    int ssd;         // объем SSD в Гбайт
    float price;     // цена
};
```

В3. ФИНАНСОВЫЕ ДОКУМЕНТЫ БУХГАЛТЕРИИ

Входной файл содержит информацию о работниках некоторой организации:

```

struct Rec1
{
    long nomer;           // номер записи в файле
    char name[20];       // фамилия
    int year;            // год рождения
    float zarpl;        // заработная плата
};

```

Требуется в выходной файл записать все фамилии работников без повторения в отсортированном виде:

```

struct Rec2
{
    long nomer;           // номер записи в файле
    char name[20];       // фамилия
};

```

В4. СТУДЕНЧЕСКАЯ ГРУППА

Входной файл содержит сведения о студентах некоторой группы:

```

struct Rec1
{
    long nomer;           //номер записи в файле
    char name[30];       //фамилия
    int year;            //год рождения
    float sred;         //средний балл
};

```

Требуется в выходной файл записать сведения о тех студентах, которые имеют наибольший средний балл:

```

struct Rec2
{
    char name[20];       // фамилия
    int year;            // год рождения
    float sred;         // средний балл
};

```

В5. СТУДЕНТЫ ФАКУЛЬТЕТА

Входной файл содержит информацию о студентах некоторого факультета:

```
struct Rec1
{
    long nomer;           // номер записи в файле
    char name[30];       // фамилия
    int year;            // год рождения
    char group[30];      // группа
    float sred;         // средний балл
};
```

Требуется в выходной файл записать информацию о студентах, которые принадлежат какой-то одной конкретной группе, причем записи должны быть отсортированы в алфавитном порядке относительно фамилий:

```
struct Rec2
{
    long nomer;           // номер записи в файле
    char name[20];       // фамилия
    int year;            // год рождения
    float sred;         // средний балл
};
```

В6. РАБОТА С ФАЙЛАМИ

Во входном файле задана последовательность записей, которые содержат полный путь к тому или иному файлу на жестком диске:

```
struct Rec1
{
    long nomer;           // номер записи в файле
    char fileName[256]; // полный путь к файлу
};
```

Требуется из полного имени файла `fileName` выделить только имя файла и записать его в переменную `name`. При считывании очередной записи из входного файла требуется проверить, имеется ли файл с именем `fileName` на жестком диске. Если такой файл имеется, то в переменную `flag` записать 1, иначе 0. Также в переменную `size` записать размерность файла

(в байтах); если же такого файла нет, то в переменную size записать -1.

Компоненты выходного файла имеют такой вид:

```
struct Rec2
{
    long nomer;           // номер записи в файле
    char fileName[256]; // полный путь к файлу
    char name[256];      // имя файла без учета каталогов
    int flag;            // наличие файла с именем fileName на жестком
диске
    long size;          // размерность файла с именем fileName в байтах
};
```

В7. ГОРОДА

Входной файл содержит информацию о городах:

```
struct Rec1
{
    long nomer;           // номер записи в файле
    char name[50];       // название
    int year;            // год основания
    long count;         // количество жителей
};
```

Требуется в выходной файл записать информацию о городах, которые являются самыми густонаселенными:

```
struct Rec2
{
    char name[50];       // название
    int year;            // год основания
    long count;         // количество жителей
};
```

В8. МАГАЗИН

Входной файл содержит информацию о товарах:

```
struct Rec1
{
    long nomer;           // номер записи в файле
```

```

char name[50];    // название
char firm[100];  // фирма-изготовитель
char date[50];   // дата изготовления
float price;     // цена
};

```

Требуется в выходной файл записать сведения о тех продуктах, которые являются самыми дорогими:

```

struct Rec2
{
    char name[50];    // название
    char firm[100];  // фирма-изготовитель
    char date[50];   // дата изготовления
    float price;     // цена
};

```

В9. БИБЛИОТЕКА

Входной файл содержит сведения о книгах некоторой библиотеки:

```

struct Rec1
{
    int nomer;       // номер записи в файле
    char name[30];   // фамилия автора
    char nazv[30];   // название книги
    int year;        // год издания
};

```

Требуется в выходной файл записать данные о тех книгах, которые являются самыми старыми:

```

struct Rec2
{
    char name[30];   // фамилия автора
    char nazv[30];   // название книги
    int year;        // год издания
};

```

В10. ОРГАНИЗАЦИЯ

Входной файл содержит информацию о работниках некоторой организации:

```
struct Rec1
{
    long nomer;           // номер записи в файле
    char name[20];       // фамилия
    int year;            // год рождения
    float zarpl;         // заработная плата
};
```

Требуется в выходной файл записать сведения о тех работниках, которые являются самыми молодыми:

```
struct Rec2
{
    char name[20];       // фамилия
    int year;            // год рождения
    float zarpl;         // заработная плата
};
```

В11. СТУДЕНТЫ ФАКУЛЬТЕТА

Входной файл содержит информацию о студентах некоторого факультета:

```
struct Rec1
{
    long nomer;           // номер записи в файле
    char name[30];       // фамилия
    int year;            // год рождения
    char group[30];     // группа
    float sred;          // средний балл
};
```

Требуется в выходной файл записать все сведения о студентах из одной фиксированной группы:

```
struct Rec2
{
    char name[30];       // фамилия
```

```
int year;           // год рождения
float sred;        // средний балл
};
```

В12. СТРАНЫ

Входной файл содержит информацию о странах:

```
struct Rec1
{
    long nomer;           // номер записи в файле
    char name[50];       // название
    char language[50];  // государственный язык
    float size;          // площадь
    long count;         // количество жителей
};
```

Требуется в выходной файл записать информацию о таких странах, которые имеют наибольшую площадь:

```
struct Rec2
{
    char name[50];       // название
    float size;          // площадь
    long count;         // количество жителей
};
```

В13. ПЕРЕВОД ЧИСЕЛ В ДРУГУЮ СИСТЕМУ СЧИСЛЕНИЯ

Входной файл содержит структуры:

```
struct Rec1
{
    long nomer;           // номер записи в файле
    long a;              // число в 10-тичной системе
    unsigned char basis; // основание новой системы счисления
};
```

Во входном файле задана последовательность десятичных чисел. Получить последовательность тех же чисел, но в системе счисления, определяемой полем «basis» (значение которого может быть от 2 до 36), и в

символьном представлении. Дополнительные цифры отобразить буквами «А», «В», «С», «D» и т. д.

Полученные числа записать в выходной файл (FOut) типа:

```
struct Rec2
{
    long nomer;           // номер записи в файле
    unsigned char basis; // основание новой системы счисления
    char rez[256];       // число в соответствующей системе счисления
};
```

В14. КАТАЛОГ МОНИТОРОВ

Входной файл содержит сведения о мониторах LCD:

```
struct Rec1
{
    int nomer;           // номер записи в файле
    char mark[20];      // производитель
    int diag;           // диагональ в дюймах
    char matrix[20];    // тип матрицы (TN, IPS, PLS и т.д.)
    float price;        // цена
};
```

Требуется в выходной файл записать данные о тех мониторах, которые имеют наибольшую диагональ:

```
struct Rec2
{
    char mark[20];      // производитель
    int diag;           // диагональ в дюймах
    char matrix[20];    // тип матрицы
    float price;        // цена
};
```

В15. ИНФОРМАЦИЯ О ФАЙЛАХ

Входной файл содержит сведения о некоторых файлах на ПК:

```
struct Rec1
{
    long nomer;         // номер записи в файле
```

```
char fileName[256]; // полный путь к файлу
long size;          // размерность файла с именем fileName в байтах
};
```

Требуется в выходной файл записать информацию о тех файлах, которые имеют наибольший размер:

```
struct Rec2
{
    char fileName[256]; // полный путь к файлу
    long size;          // размерность файла с именем fileName в байтах
};
```

Уровень С

Разработать программу, основной блок которой в режиме «МЕНЮ» позволяет выбрать и запустить рабочие функции следующего назначения:

- CreateFileIn(char *fileName) – создание двоичного файла данных, элементами которого являются записи со структурой, указанной в индивидуальном варианте.
- AddFileIn (char *filename) – добавление записей в конец входного файла.
- ReadFileIn(char *fileName) – просмотр записей во входном файле последовательно от первой до последней.
- CreateFileOut(char *fileName1, char *fileName2) – обработка данных из входного файла по заданию индивидуального варианта и вывод результатов в выходной файл.
- ReadFileOut(char *fileName) – просмотр записей в выходном файле последовательно от первой до последней.
- Две функции сортировки, которые сортируют входной файл по двум полям (поля выбрать по желанию).

Варианты заданий

С1. САМОЛЕТЫ

Входной файл содержит информацию о самолетах:

```
struct Rec1
{
    int nomer;          // номер записи в файле
```

```

char name[50];    // название самолета
int year;        // год выпуска
float weight;    // грузоподъемность в тоннах
float price;     // цена
};

```

Требуется в выходной файл записать данные о тех самолетах, которые среди самых старых имеют наибольшую грузоподъемность:

```

struct Rec2
{
    char name[50];    // название самолета
    int year;        // год выпуска
    float weight;    // грузоподъемность в тоннах
    float price;     // цена
};

```

С2. КАТАЛОГ ПРОЦЕССОРОВ

Входной файл содержит сведения о процессорах:

```

struct Rec1
{
    int nomer;       // номер записи в файле
    char mark[20];   // производитель
    char core[20];   // ядро процессора
    int frequency;   // частота
    int count;       // количество ядер
    float price;     // цена
};

```

Требуется в выходной файл записать данные о тех процессорах, которые среди самых дорогих имеют наибольшую частоту:

```

struct Rec2
{
    char mark[20];   // производитель
    char core[20];   // ядро процессора
    int frequency;   // частота
    float price;     // цена
};

```

С3. ХАРАКТЕРИСТИКИ НОУТБУКОВ

Входной файл содержит сведения о ноутбуках:

```
struct Rec1
{
    int nomer;           // номер записи в файле
    char name[30];      // производитель
    int diag;           // диагональ в дюймах
    int frecuency;      // тактовая частота
    int ddr;            // объем ОП в Мбайт
    int hdd;            // объем HDD или SSD в Гбайт
    float price;        // цена
};
```

Требуется в выходной файл записать информацию о тех ноутбуках, которые среди самых дорогих имеют наибольшую диагональ:

```
struct Rec2
{
    char name[30];      // производитель
    int diag;           // диагональ в дюймах
    float price;        // цена
};
```

С4. КАТАЛОГ МОНИТОРОВ

Входной файл содержит сведения о мониторах LCD:

```
struct Rec1
{
    int nomer;           // номер записи в файле
    char mark[20];       // производитель
    int diag;           // диагональ в дюймах
    char matrix[20];     // тип матрицы
    float price;        // цена
};
```

Требуется в выходной файл записать информацию о тех мониторах, которые среди самых дорогих имеют наименьшую диагональ:

```
struct Rec2
{
    char mark[20];       // производитель
```

```
int diag;           // диагональ в дюймах
char matrix[20];   // тип матрицы
float price;       // цена
};
```

С5. ФИНАНСОВЫЕ ДОКУМЕНТЫ БУХГАЛТЕРИИ

Входной файл содержит информацию о работниках некоторой организации:

```
struct Rec1
{
    long nomer;       // номер записи в файле
    char name[20];    // фамилия
    int year;         // год рождения
    float zarpl;     // заработная плата
};
```

Требуется в выходной файл записать сведения о тех работниках, которые среди самых молодых имеют наибольшую заработную плату:

```
struct Rec2
{
    char name[20];    // фамилия
    int year;         // год рождения
    float zarpl;     // заработная плата
};
```

С6. КНИЖНЫЙ МАГАЗИН

Входной файл содержит информацию о книгах:

```
struct Rec1
{
    int nomer;        // номер записи в файле
    char name[20];    // фамилия автора
    char nazv[30];    // название книги
    int year;         // год издания
    float price;     // цена
};
```

Требуется в выходной файл записать сведения о тех книгах, которые среди самых старых являются самыми дешевыми:

```
struct Rec2
{
    char name[20];    // фамилия автора
    char nazv[30];   // название книги
    int year;        // год издания
    float price;     // цена
};
```

С7. КАТАЛОГ СОТОВЫХ ТЕЛЕФОНОВ

Входной файл содержит информацию о сотовых телефонах:

```
struct Rec1
{
    long nomer;      // номер записи в файле
    char name[20];   // производитель
    int diag;        // диагональ в дюймах
    int mem;         // объем памяти
    float price;     // цена
};
```

Требуется в выходной файл записать информацию о тех телефонах, которые среди самых дорогих имеют наибольшую диагональ:

```
struct Rec2
{
    char name[20];   // производитель
    int diag;        // диагональ в дюймах
    float price;     // цена
};
```

С8. СТУДЕНТЫ ФАКУЛЬТЕТА

Входной файл содержит информацию о студентах некоторого факультета:

```
struct Rec1
{
    long nomer;      // номер записи в файле
```

```
char name[20];    // фамилия
int year;        // год рождения
char group[20];  // группа
float sred;      // средний балл
};
```

Требуется в выходной файл записать сведения о тех студентах, которые среди самых молодых имеют максимальный средний балл:

```
struct Rec2
{
    char name[20];    // фамилия
    int year;        // год рождения
    char group[20];  // группа
    float sred;      // средний балл
};
```

Лабораторная работа № 12. Линейные списки

Уровень А

Требуется создать линейный список целых чисел, элементами которого являются случайные целые числа или вводимые с клавиатуры, и определить его характеристики в соответствии с вариантом.

Программа должна содержать следующие функции:

- создание списка и заполнение его элементов случайными числами или вводимыми с клавиатуры;
- вывод списка на экран;
- удаление всего списка.

Варианты заданий

A1. В линейном списке целых чисел найти максимум и минимум.

A2. Найти количество и сумму тех элементов линейного списка, которые делятся на 5 и не делятся на 7.

A3. Найти количество простых чисел в линейном списке.

A4. Найти количество и сумму положительных элементов линейного списка, которые делятся на 5.

A5. Найти количество составных чисел в линейном списке.

A6. Вычислить сумму только симметричных элементов линейного списка.

A7. Вычислить среднее арифметическое чисел линейного списка, содержащих только четные цифры.

A8. Найти количество элементов линейного списка, больших предыдущего и меньших последующих.

A9. В линейном списке целых чисел найти количество минимальных элементов.

A10. Вычислить сумму элементов линейного списка, являющихся простыми числами и расположенных в интервале (A, B).

A11. Найти количество элементов линейного списка, больших всех предыдущих элементов.

A12. В линейном списке целых чисел найти количество максимальных элементов.

A13. Найти сумму всех двузначных элементов линейного списка.

A14. Найти количество таких элементов линейного списка, которые имеют в младшем разряде цифру 7.

A15. Вычислить сумму элементов линейного списка, содержащих хотя бы одну нечетную цифру.

A16. В линейном списке целых чисел найти количество простых чисел.

A17. В линейном списке целых чисел найти среднее арифметическое нечетных чисел, делящихся на 5.

Уровень В

Имеется текстовый файл с целыми числами. Требуется сначала все числа из данного файла записать в линейный список. Затем найти количество элементов в линейном списке, обладающих свойством Q. После чего удалить из списка все элементы, обладающие свойством T. Свойства Q и T указаны в индивидуальных вариантах. После каждого преобразования линейного списка вывести его на экран (после заполнения и после удаления элементов).

Программа должна содержать следующие функции:

- создание списка и заполнение его целыми числами из текстового файла;
- вывод списка на экран;
- количество элементов списка, обладающих свойством Q;
- удаление всех элементов списка, обладающих свойством T;
- удаление всего списка.

Варианты заданий

V1. Q: число является четным. T: число содержит в своем составе цифры 2 и 3.

V2. Q: число является положительным. T: число начинается с цифры 1.

V3. Q: число содержит цифру 1. T: число является простым.

- В4. Q: число не содержит цифру 0. T: первая цифра числа совпадает с последней.
- В5. Q: число является нечетным. T: число является составным.
- В6. Q: число является простым. T: число содержит в своем составе цифру 5.
- В7. Q: число является двузначным. T: число не содержит в своем составе цифру 7.
- В8. Q: число является симметричным. T: число является трехзначным.
- В9. Q: сумма цифр числа равна 10. T: число является четным.
- В10. Q: число не содержит четных цифр. T: число является простым.
- В11. Q: число является составным. T: число начинается с 2.
- В12. Q: число начинается с 3. T: число является составным.
- В13. Q: число является степенью двойки. T: первая цифра числа является нечетной.
- В14. Q: число содержит хотя бы одну четную цифру. T: число содержит в младшем разряде цифру 3.
- В15. Q: число содержит цифру 3. T: число является составным.
- В16. Q: число является трехзначным. T: число содержит в своем составе цифру 2.
- В17. Q: число является степенью тройки. T: число не содержит в своем составе цифру 5.

Уровень С

Имеется текстовый файл с целыми числами. Требуется сначала все числа из данного файла записать в двусвязный линейный список. Затем удалить из списка все элементы, обладающие свойством Q. После чего продублировать все элементы линейного списка, обладающие свойством R. Наконец, упорядочить все элементы линейного списка, обладающие свойством T, оставив остальные элементы на своих местах. Свойства Q, R и T указаны в индивидуальных вариантах. После каждого преобразования линейного списка вывести его на экран.

Программа должна содержать следующие функции:

- создание списка и заполнение его целыми числами из текстового файла;
- вывод списка на экран;
- удаление всех элементов списка, обладающих свойством Q;
- дублирование всех элементов списка, обладающих свойством R;
- сортировка всех элементов списка, обладающих свойством T;
- удаление всего списка.

Варианты заданий

C1. Q: число является отрицательным. R: число не содержит четных цифр. T: число начинается с цифры 1.

C2. Q: число содержит цифру 1. R: число является составным. T: число является простым.

C3. Q: число не содержит цифру 0. R: число начинается с 2. T: первая цифра числа совпадает с последней.

C4. Q: число является нечетным. R: число начинается с 3. T: число является составным.

C5. Q: число является простым. R: число является степенью двойки. T: число содержит в своем составе цифру 5.

C6. Q: число является двузначным. R: число является четным. T: число не содержит в своем составе цифру 7.

C7. Q: число является симметричным. R: число содержит хотя бы одну четную цифру. T: число является трехзначным.

C8. Q: сумма цифр числа равна 10. R: число содержит в младшем разряде цифру 3. T: число является двузначным.

C9. Q: сумма цифр числа является четным числом. R: число является произведением двух различных простых чисел. T: число является простым.

C10. Q: число является квадратом некоторого натурального числа. R: число является степенью некоторого простого числа. T: число является составным.

Лабораторная работа № 13. Обработка линейных динамических структур

Данная лабораторная работа является продолжением лабораторной работы № 11. **Выполнить тот же вариант, что и в лабораторной работе № 11.**

Уровень А

Требуется создать двоичный файл, выгрузить все его содержимое в линейный список и обработать список в соответствии с вариантом.

Уровни В и С

Требуется создать двоичный файл Fin, выгрузить все его содержимое в линейный список LIn. Выполнить тот же вариант, что и в лабораторной работе № 11, но при этом сначала создать результирующий список LOut, а уже после этого информацию из LOut переместить в FOut. Схема данной лабораторной работы такая:

Fin -> LIn -> LOut -> FOut.

Все операции необходимо выполнять **только** со списком LIn, файл Fin нужен только для хранения информации. Из него разрешается только считать информацию в LIn.

Лабораторная работа № 14. Обработка бинарных деревьев

Уровень В

Имеется текстовый файл с изображением целых чисел. Требуется сначала все числа из данного файла записать в двоичное дерево. Затем обработать дерево в соответствии с заданием.

Программа должна содержать следующие функции.

- `CreateTree(char *fname, ELEMENT **root)` – построение дерева из элементов файла `fname`.
- `PrintTree(ELEMENT *root)` – вывод дерева на экран.
- `DestroyTree(ELEMENT **root)` – удаление дерева.

Варианты заданий

В1. Дан текстовый файл с изображениями целых чисел. Построить из них сбалансированное дерево. Определить количество элементов в левом поддереве той вершины, которая является максимальным элементом в дереве.

В2. Дан текстовый файл с изображением целых чисел. Построить сбалансированное дерево. Определить количество элементов на самом нижнем уровне дерева.

В3. Дан текстовый файл с изображением целых чисел. Построить дерево поиска. Определить уровень дерева, на котором все элементы являются четными. Найти их сумму и количество листьев на этом уровне.

В4. Дан текстовый файл с изображением целых чисел. Построить сбалансированное дерево, не включая в него повторно встреченные элементы. Определить количество элементов-листьев в левом поддереве вершины, которая имеет минимальное значение.

В5. Дан текстовый файл с изображением целых чисел. Построить дерево поиска. Определить количество элементов на самом нижнем уровне дерева.

В6. Дан текстовый файл с изображением целых чисел. Построить дерево поиска. Удалить из дерева все вершины, содержащие простые числа.

В7. Дан текстовый файл с изображением целых чисел. Построить дерево поиска. Определить уровень, на котором находится наибольшее количество вершин-листьев.

В8. Дан текстовый файл с изображением целых чисел. Построить сбалансированное дерево. Определить количество наиболее часто встречающегося элемента в дереве.

В9. Дан текстовый файл с изображением целых чисел. Построить сбалансированное дерево. Определить количество простых чисел на самом нижнем уровне дерева.

В10. Дан текстовый файл с изображением целых чисел. Построить дерево поиска. Определить количество вершин в дереве, имеющих ровно двух потомков.

В11. Дан текстовый файл с изображением целых чисел. Построить сбалансированное дерево, не включая в него повторно встреченные элементы. Определить количество вершин в дереве, имеющих только одного потомка.

В12. Дан текстовый файл с изображениями целых чисел. Построить сбалансированное дерево, не включая в него повторно встреченные элементы. Определить количество элементов в правом поддереве той вершины, которая является минимальным элементом в дереве.

В13. Дан текстовый файл с изображением целых чисел. Построить дерево поиска. Определить количество элементов на каждом четном уровне дерева.

В14. Дан текстовый файл с изображением целых чисел. Построить дерево поиска. Определить количество всех простых элементов в дереве.

В15. Дан текстовый файл. Построить частотный словарь слов на основе дерева поиска. Частотный словарь (слова которого лексикографически упорядочены) записать в другой текстовый файл.

Уровень С

Имеется текстовый файл с изображением целых чисел. Создать два бинарных дерева из данных чисел: сбалансированное дерево T1 и дерево поиска T2. На каждом уровне дерева T1 найти минимальный из всех элементов, которые обладают свойством Q. Также вывести на экран все пути

в дереве T_2 , ведущие от корня к листьям, содержащие хотя бы один элемент, обладающий свойством R .

Программа должна содержать следующие функции:

- построение сбалансированного дерева из чисел файла;
- построение дерева поиска из чисел файла;
- вывод произвольного бинарного дерева на экран;
- нахождение минимальных элементов на каждом уровне и запись их в массив;
- вывод путей от корня к листьям, отвечающих заданным условиям;
- удаление произвольного бинарного дерева.

Варианты заданий

C1. Q: число является четным. R: число состоит из попарно различных цифр.

C2. Q: число является положительным. R: только одна цифра в числе встречается ровно два раза, остальные – не более одного раза.

C3. Q: число содержит цифру 1. R: цифры в числе упорядочены по возрастанию.

C4. Q: число не содержит цифру 0. R: число является симметричными в двоичном представлении.

C5. Q: число является нечетным. R: число является простым и не содержит в своем составе цифру 5.

C6. Q: число является простым. R: в числе хотя бы одна цифра встречается более одного раза.

C7. Q: число является двузначным. R: число состоит только из простых цифр.

C8. Q: число начинается с 1. R: каждая цифра, входящая в число, встречается ровно два раза.

C9. Q: сумма цифр числа является четным числом. R: число является произведением двух различных простых чисел.

C10. Q: число является квадратом некоторого натурального числа. R: число является степенью некоторого простого числа.

Лабораторная работа № 15. Работа с битами памяти

Уровень С

Пусть имеется некоторое n -элементное множество целых чисел M ($10 \leq n \leq 20$), элементы которого записаны в массив размера n (элементы массива попарно различны). Требуется в текстовый файл записать построчно все такие подмножества множества M , каждое из которых содержит хотя бы один элемент, обладающий свойством Q . Свойства Q указаны в индивидуальных вариантах.

Варианты заданий

- С1. Q : число состоит из попарно различных цифр.
- С2. Q : четные цифры в числе чередуются с нечетными.
- С3. Q : цифры в числе упорядочены по возрастанию.
- С4. Q : число является симметричным в двоичном представлении.
- С5. Q : число является простым и не содержит в своем составе цифру 5.
- С6. Q : в числе хотя бы одна цифра встречается более одного раза.
- С7. Q : число состоит только из простых цифр.
- С8. Q : число является составным.
- С9. Q : число является произведением двух различных простых чисел.
- С10. Q : число является степенью некоторого простого числа.

Литература

1. Дейтел Х. М. Как программировать на С / Х. М. Дейтел, П. Дж. Дейтел. – М. : Бином-Пресс, 2009. – 910 с.
2. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений / Р. Хэзфилд [и др.]. – М. : ДиаСофт, 2001. – 736 с.
3. Керниган Б. Язык программирования Си / Б. Керниган, Д. Ритчи. – М. : Издательский дом «Вильямс», 2009. – 304 с.
4. Подбельский В.В. Программирование на языке Си / В. В. Подбельский, С. С. Фомин. – М. : Финансы и статистика, 2004. – 600 с.
5. Прата С. Язык программирования С : лекции и упражнения / С. Прата. – М. : Издательский дом «Вильямс», 2006. – 960 с.
6. Рацеев С. М. Задачи по программированию и основные алгоритмы. Ч. 1, 2 / С. М. Рацеев. – Ульяновск : УлГУ, 2009.
7. Рацеев С. М. Язык Си. Структуры данных и алгоритмы / С. М. Рацеев. – Ульяновск : УлГУ, 2011. – 214 с.
8. Рацеев С. М. Программирование на языке Си [Электронный учебный курс] / С. М. Рацеев. – Ульяновск : УлГУ, 2012. – 1 CD-R. – № гос. регистрации 0321301786.
9. Скляр В. А. Программирование на языках Си и Си++ / В. А. Скляр. – М. : Высшая школа, 1999. – 288 с.
10. Угаров В. В. Информатика и программирование / В. В. Угаров, Ю. В. Цыганова. – Ульяновск : УлГУ, 2003. – 62 с.
11. Шень А. Программирование: теоремы и задачи / А. Шень. – М. : МЦНМО, 2004.

Учебное издание

Сергей Михайлович Рацев

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ

Директор Издательского центра *Т.В. Филиппова*
Редактирование, подготовка оригинал-макета *Е.П. Мамаевой*
Оформление обложки *Н.В. Пеньковой*

Подписано в печать 20.08.2014. Формат 60×84/16
Гарнитура Times New Roman. Усл. печ. л. 5,3. Уч.-изд. л. 4,6
Тираж 100 экз. Заказ № 82/

Оригинал-макет подготовлен в Издательском центре
Ульяновского государственного университета
432017, г. Ульяновск, ул. Л. Толстого, 42

Отпечатано с оригинал-макета в Издательском центре
Ульяновского государственного университета
432017, г. Ульяновск, ул. Л. Толстого, 42