

ГЛАВА 3

ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ, ОПРЕДЕЛЁННЫХ ПОЛЬЗОВАТЕЛЕМ

ВВЕДЕНИЕ

В задачах этого раздела требуется логически независимые или повторяющиеся последовательности действий оформить в виде отдельных функций, к которым обращаться из функции *main()*. В задачах, работающих с массивами, необходимо предусмотреть возможность ручного ввода массива, а также заполнение его с помощью датчика случайных чисел.

Каждый студент должен решить три задачи, начиная с задачи, номер которой вычисляется по формуле:

$$НЗ = (2 * N) \% 63, \text{ где } N - \text{ порядковый номер студента.}$$

Две задачи, по указанию преподавателя, отладить на ЭВМ.

ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ

Задача А

Написать и протестировать функцию для вычисления x^n .

```
#include <stdio.h>

void main()
{
    int n=-4;
    float power(float, int), x=2.0;

    printf("-3 в степени 3 =%f\n", power(-3.0, 3));
    printf("%4.2f в степени %3d =%6.2f\n", x, n, power(x, n));
    printf("Возводим 0 в степень -2:\n");
    power(0.0, -2);
}

float power (float x, int n)
{
    float step;

    if(n>0)
    {
        for(step=1.0; n>0; n--) step*=x;
        return step;
    }
    else if(x!=0)
    { for( step=1.0; n<0; n++) step/=x;
      return step;
    }
    else
    {
        puts("Ошибка!!!");
    }
}
```

```

        printf("Нельзя возводить 0 в %d степень!\n",n);
        return 0;
    }
}

```

Функция *power()* должна вернуть в вызывающую функцию одно значение. Для этого удобно использовать оператор *return*. Значение, возвращаемое таким образом, является результатом работы функции и может использоваться в выражениях. Аргументы функции не должны изменяться после ее работы, поэтому они передаются в функцию по значению.

Чтобы проверить правильность работы написанной функции, необходимо протестировать ее на нескольких типах входных данных. Тестовые данные надо подобрать таким образом, чтобы проследить работу всех ветвей составленного алгоритма. В нашем случае достаточно посмотреть, как работает функция вычисления степени, на примере данных трех типов, что и сделано в функции *main()*.

Задача В

Написать и протестировать функцию, которая "сжимает" строку, удаляя из нее каждый второй элемент.

```

#include <stdio.h>

void main()
{
    void squeeze(char *);
    char s[80], answer;

    do
    {
        puts("\nВведите строку символов");
        gets(s);
        puts("\nВот сжатая строка:");
        squeeze(s);
        puts(s);
        puts("\n\nПопробуем еще раз? (Y/N)");
        answer=getchar(); getchar();
    }
    while (answer=='Y' || answer=='y');
}

void squeeze(char str[])
{
    int i=-1, k=0;

    while (str[++i])
        if (i%2==0) str[k++]=str[i];
    str[k]='\0';
}

```

Можно предложить другой вариант функции *squeeze()*, например, реализовать ее с использованием указателей. При этом в работе пользователя никаких изменений не произойдет, т.е. функция *main()* останется прежней.

Массив передается в функцию по адресу, поэтому не приходится заботиться о том, чтобы возвращать измененные значения его элементов.

Задача С

Написать и протестировать функцию $zam(x, y)$, которая свой первый параметр заменяет на сумму, а второй - на разность аргументов.

```
#include <stdio.h>

void main()
{
    void zam(int *, int *);
    int a=5, b=15, c=20, d=-3;

    printf("a=%d, b=%d\nc=%d, d=%d\n", a, b, c, d);
    zam(&a, &b);
    zam(&c, &d);
    puts("\nА теперь -");
    printf("a=%d, b=%d\nc=%d, d=%d\n", a, b, c, d);
}

void zam (int *x, int *y)
{
    int buf;

    buf=*x;
    *x=*x+*y;
    *y=buf-*y;
}
```

В данном примере функция $zam()$ должна вернуть сразу два значения, поэтому оператором *return* тут уже не обойтись. Кроме того, с помощью функции $zam()$ мы собираемся изменить значения переменных, которые принадлежат другой функции. Поэтому в функцию $zam()$ приходится передавать *адреса* переменных, т.е. аргументы функции должны быть *указателями* соответствующего типа, а в самой функции работа будет идти с *содержимым* этих адресов. Благодаря передаче аргументов *по адресу*, а не *по значению*, функция получает доступ к "чужим" ячейкам памяти.

Задача D

Дано натуральное число $n > 13$. Выдать все пары "близнецов", меньших n ("близнецы" - простые числа, разность между которыми равна 2).

```
include <math.h>
include <stdio.h>

void main()
{
    int n, i, i1;
    int simple (int);

    do
    {
        puts("Введите натуральное N");
        scanf("%d", &n);
    }
    while (n<=0);
    printf("Пары \"близнецов\", меньших %d:\n", n);
    for (i=1; i<n-2; i+=2) // среди простых чисел нет четных
```

```

{
    i1=i+2;           // "близнецы" отличаются друг от друга на 2
    if (simple(i) && simple(i1))
        printf("%d\t%d\n", i, i1);
}
}

int simple (int x)
{
int i;

    if(x==1) return 1;
    for (i=2; i<sqrt(x); i++) // перебираем возможные делители числа x
        if(x%i==0) return 0; // если нашелся делитель - число x не простое
    return 1; // делителей не было
}

```

Задача E

Написать и протестировать функцию для перемножения двух квадратных матриц.

```

#include <stdlib.h>
#include <stdio.h>
#define N 5
#define rnd(a, b) (a+(b-a)*rand()/32768.0)
#define MEMORY (float *)malloc(N*N*sizeof(float))

// функция mult() - перемножение матриц
void mult(float *a, int n, int m, float *b, int l, float *c)
{
    int i, j, k;
    float sum;

    for(i=0; i<n; i++)
        for(j=0; j<l; j++)
            {
                for(sum=k=0; k<m; k++)
                    sum+=*(a+i*m+k)**(b+k*l+j);
                *(c+i*l+j)=sum;
            }
}

// функция print_matr() - печать матриц
void print_matr (float *a, int n, int m)
{
    int i, j;

    for (i=0; i<n; putchar('\n'), i++)
        for (j=0; j<m; j++)
            printf("%6.3f ", *(a+i*m+j));
    putchar('\n');
}

void main()
{
    float *x, *y, *z;
    int i, j;

```

```

x=MEMORY;          y=MEMORY;  z=MEMORY;
puts("Будете вводить матрицы вручную? Да-введи 'y' : ");
if (getchar()=='y')
{
    puts("\nВводите матрицу X построчно :\n");
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            scanf("%f", x+i*N+j);
    puts("\nВводите матрицу Y построчно :\n");
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            scanf("%f", y+i*N+j);
}
else
{
    srand(1135);
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
        {
            *(x+i*N+j) = rnd(1, 5);
            *(y+i*N+j) = rnd(1, 5);
        }
}
puts("\nИсходные матрицы:");
print_matr(x, N, N); print_matr(y, N, N);
mult(x, N, N, y, N, z);
puts("\nРезультирующая матрица:");
print_matr(z, N, N);
}

```

В программе предусмотрена возможность ручного ввода матриц, который может понадобиться на этапе отладки программы. По желанию пользователя матрицы могут быть сформированы и с помощью датчика случайных чисел; в этом случае элементами матриц являются случайные числа из диапазона (1,5). Датчик для получения таких чисел оформлен в виде макро с параметрами и использует стандартную функцию *rand()*, вырабатывающую случайное целое число из диапазона (0,32767).

Задача F

Написать и протестировать функцию для приближённого вычисления $y = \arctg(x)$ с помощью многочлена наилучшего приближения

$$y \approx \sum_{k=0}^3 a_{2k+1} x^{2k+1}, \quad 0 \leq x \leq 1,$$

где $a_1 = 0.9992$, $a_3 = -0.3212$, $a_5 = 0.1463$, $a_7 = -0.0390$.

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define RND (rand()/32768.0*0.1)
float p_atan(float);
void main()

```

```

{
  int i;
  float x, atan_p, atan_t, d_atan;
  printf("\n\n Тестирование функции p_atan() ");
  for(i=0; i<5; i++)
  {
    if(i<3) x=0.1*0.45-RND;
    else x=10*RND;
    atan_p=p_atan(x);
    atan_t=atan(x);
    d_atan=atan_p-atan_t;
    printf("\n x=%.3f atan_p=%.4f atan_t=%.4f d_atan=5%e ",
           x, atan_p, atan_t, d_atan);
  }
}
*/
    Функция p_atan() служит для приближённого вычисления  $y=\arctg(x)$  на интервале (0, 1) с помощью многочлена наилучшего приближения. Квадратическая норма ошибки  $r=8e-5$ .
*/
float p_atan(float x)
{
  int k;
  static float a[]={0.9992, -0.3212, 0.1463, -0.0390};
  float sum=0, x2;

  x2=x*x;
  for(k=0 k<4; k++0
  {
    sum+=a[k]*x;
    x*=x2;
  }
  return sum;
}

```

Тестируем функцию *p_atan()* для пяти случайных значений аргумента. Первые три - из интервалов (0.0, 0.1), (0.45, 0.55), (0.9, 1.0) соответственно, а два других - из интервала (0.0, 1.0). Поскольку тестируемые точки достаточно полно охватывают диапазон возможных значений аргумента, то по поведению ошибки вычисления приближённого значения функции $\arctg(x)$ можно будет сделать вполне обоснованный вывод о правильной или ошибочной работе функции *p_atan()*.

Коэффициенты *a[k]* объявлены *static float*, чтобы избежать их многократной инициализации при повторных обращениях к функции.

ЗАДАНИЕ НА ПРОГРАММИРОВАНИЕ

Задача 1

Написать и протестировать функцию, которая "переворачивает" строку, передаваемую ей в качестве параметра.

Задача 2

Дано натуральное число n . Распечатать число, которое получится после выписывания цифр числа n в обратном порядке. (Для получения нового числа составить функцию).

Задача 3

Написать и протестировать функцию, подсчитывающую количество минимальных элементов в целочисленной матрице.

Задача 4

Написать и протестировать функцию для вычисления числа сочетаний по формуле

$$C_n^k = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\dots(n-k+1)}{k!}.$$

Задача 5

По заданным значениям $X[20]$, $Y[20]$ вычислить

$$u = \begin{cases} \sum_{i=0}^{19} x_i^2 & \text{при } \sum_{i=1}^{15} x_i y_i > 0, \\ \sum_{i=10}^{19} y_i^2 & \text{при } \sum_{i=1}^{15} x_i y_i \leq 0. \end{cases}$$

Задача 6

Написать и протестировать функцию, которая преобразует строку 8-х цифр в эквивалентное ей целое 10-е число.

Задача 7

Описать функцию $\min\max(x, y)$, которая присваивает первому параметру большее, а второму - меньшее из значений x и y . Используя эту функцию, перераспределить введенные значения переменных A , B , C так, чтобы стало $A \leq B \leq C$.

Задача 8

Даны две квадратные матрицы. напечатать ту из них, которая имеет минимальный "след" (т.е. сумму элементов главной диагонали). Использовать функцию для нахождения следа матрицы и функцию печати матрицы.

Задача 9

Составить и протестировать функцию для вычисления

$$f(x, n, m) = \sum_{i=m}^n \frac{x^{2i}}{(2i)!}.$$

Задача 10↑

Написать и протестировать функцию *compress()*, которая "сжимает" строку, удаляя из нее все пробелы.

Задача 11

Написать и протестировать функцию, которая подсчитывает, сколько раз в заданной строке встретился указанный символ.

Задача 12

Написать и протестировать функцию, которая находит в массиве минимальный по модулю элемент и заменяет им все элементы с нечетными номерами.

Задача 13

Написать и протестировать функцию, которая в прямоугольной матрице находит сумму элементов *j-ой* строки.

Задача 14

Написать и протестировать функцию, которая по заданному натуральному числу определяет количество цифр в нем и их сумму.

Задача 15

Написать и протестировать функцию, которая по заданной строке *Str* формирует новую строку, состоящую только из цифр, входящих в *Str*.

Задача 16

Написать и протестировать функцию, подсчитывающую количество положительных элементов в массиве.

Задача 17

Написать и протестировать функцию, вычисляющую $y = \sqrt[3]{x}$, используя итерационную формулу:

Начальное приближение: $y_0 = x$.

$$y_{i+1} = y_i + \frac{1}{3} \left(y_i - \frac{y_i^4}{x} \right).$$

Итерации прекратить при $|y_{i+1} - y_i| < 2 * 10^{-6}$.

Задача 18↑

Определить, является ли данная матрица ортонормированной, т.е. такой, в которой скалярное произведение каждой пары различных строк равно нулю, а скалярное произведение каждой строки на себя равно единице.

Задача 19

Составить и протестировать функцию для замены символов ':' на '.' в заданной строке, начиная с указанной позиции.

Задача 20

Выяснить, сколько простых чисел находится в интервале $[n, m]$, и распечатать их. Для определения, является ли очередное число простым, составить функцию.

Задача 21

Написать и протестировать функцию для вычисления площади треугольника, заданного координатами вершин.

Задача 22

Написать и протестировать функцию для нахождения в прямоугольной матрице номера строки, имеющей максимальную сумму элементов.

Задача 23

Написать и протестировать функцию для приближённого вычисления $\log_2 x$ с помощью многочлена наилучшего приближения

$$\log_2 x \approx \sum_{k=1}^3 a_{2k-1} \left(\frac{x-1}{x+1} \right)^{2k-1}, \quad 1 \leq x \leq \sqrt{2},$$

где $a_1 = 2.8854$, $a_3 = 0.9615$, $a_5 = 0.5990$.

Задача 24

Написать и протестировать функцию, которая преобразует строку 2-х цифр в эквивалентное ей целое 10-е число.

Задача 25

Написать и протестировать функцию, которая в строке, передаваемой ей в качестве параметра, заменяет каждый второй элемент на заданный символ.

Задача 26

Написать и протестировать функцию для сложения и вычитания вещественных матриц. Одним из формальных параметров должен быть признак вида операции.

Задача 27

Вычислить значение

$$z = \frac{v_1 + v_2 + v_3}{3},$$

где v_i - объёмы шаров с радиусами r_i . (Использовать функцию для нахождения объёма шара).

Задача 28

Написать и протестировать функцию, которая определяет, располагаются ли буквы в заданной символьной строке в алфавитном порядке.

Задача 29↑

Даны вещественные числа a, b, c, d, e, f . Переменной S присвоить значение 1, если оба уравнения: $ax^2 + bx + c = 0$ и $dx^2 + ex + f = 0$ имеют вещественные корни, и при этом все корни первого уравнения лежат между корнями второго уравнения. В противном случае переменной S присвоить значение 0. (Для нахождения корней квадратного уравнения использовать функцию).

Задача 30

Написать и протестировать функцию для приближённого вычисления $\ln x$ по формуле

$$\ln x \approx -\frac{1}{2} \ln 2 + \sum_{k=0}^2 a_{2k+1} u^{2k+1}, \quad u = \frac{2x - \sqrt{2}}{2x + \sqrt{2}}, \quad \frac{1}{2} \leq x \leq 1,$$

где $a_1 = 2.0$, $a_3 = 0.6664$, $a_5 = 0.4150$.

Задача 31

Написать и протестировать функцию, которая преобразует целое без знака в его восьмеричное символьное представление (библиотечные функции для преобразования числа в строку и формат вывода "%o" не использовать).

Задача 32

Написать и протестировать функцию, переставляющую в обратном порядке элементы главной диагонали квадратной матрицы.

Задача 33

Разработать функцию, возвращающую наименьшее общее кратное трёх заданных натуральных чисел.

Задача 34

Даны длины отрезков A, B, C, D . Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь этого треугольника. (Определить функцию для нахождения и печати площади треугольника, если он существует).

Задача 35

Составить и протестировать функцию, возвращающую номер самого правого вхождения заданного символа во введенную строку. Если символ не входит в строку, должно возвращаться -1.

Задача 36

Написать и протестировать функцию, которая определяет, входит ли каждая буква в заданную строку не более двух раз.

Задача 37

Написать и протестировать функцию для транспонирования матрицы.

Задача 38

Найти натуральное число из интервала $[n1, n2]$ с максимальной суммой делителей. (Для нахождения суммы делителей числа использовать функцию).

Задача 39

Составить и протестировать функцию для вычисления

$$f(n, m) = \frac{n! + m!}{m!(n-m)!}, \quad 0 \leq m \leq n.$$

Задача 40

Написать и протестировать функцию, которая преобразует целое без знака в его шестнадцатеричное символьное представление (библиотечные функции для преобразования числа в строку и формат вывода "%x" не использовать).

Задача 41

Площадь треугольника, заданного координатами своих вершин, находится по формуле

$$S = 0.5 * |x_1y_2 + x_2y_3 + x_3y_1 - x_1y_3 - x_2y_1 - x_3y_2|.$$

Используя функцию для вычисления площади треугольника, определить площадь выпуклого четырехугольника ABCD, заданного координатами своих вершин.

Задача 42

Написать и протестировать функцию, которая определяет, являются ли симметричными в заданной квадратной матрице главная и побочная диагонали.

Задача 43

Написать и протестировать функцию для приближённого вычисления функции $f(x) = e^{-x}$ по формуле:

$$f(x) = \frac{1}{\left(\sum_{k=0}^3 a_k x^k \right)^4},$$

где $a_0 = 1.0$, $a_1 = 0.250721$, $a_2 = 0.029273$, $a_3 = 0.003828$.

Задача 44↑

Поле шахматной доски определяется парой натуральных чисел, первое из которых задает номер вертикали, а второе - номер горизонтали. Даны натуральные числа k, l, m, n . Требуется, если возможно, с поля (k, l) одним ходом коня попасть на поле (m, n) . Если нет - то определить можно ли это сделать за два хода. В случае успеха указать промежуточное поле. Использовать функцию.

Задача 45

Написать и протестировать функцию, которая преобразует целое без знака в его двоичное символьное представление (библиотечные функции для преобразования числа в строку не использовать).

Задача 46

Написать и протестировать функцию, которая все нулевые элементы заданного массива переносит в его конец, а остальные - в начало (сохраняя порядок их следования).

Задача 47

Даны длины сторон A , B , C некоторого треугольника. Определить его медианы. Длина медианы, проведенной к стороне A , вычисляется по формуле

$$l_a = 0.5\sqrt{2B^2 + 2C^2 - A^2}.$$

(Использовать функцию для нахождения длины медианы).

Задача 48

Написать и протестировать функцию, которая определяет, является ли симметричной заданная символьная строка.

Задача 49

Написать и протестировать функцию, которая по заданному натуральному числу определяет его первую и последнюю цифры.

Задача 50

Написать и протестировать функцию, которая по натуральному k и вещественным $x > 0$, $0 < \varepsilon \leq 10^{-6}$ вычисляет значение $\sqrt[k]{x}$, используя следующую итерационную формулу:

$$y_0 = x, \quad y_{i+1} = y_i + \frac{1}{k} \left(\frac{x}{y_i^{k-1}} - y_i \right).$$

В качестве результата берется то значение y_{i+1} , для которого $|y_{i+1} - y_i| < \varepsilon$.

Задача 51

Написать и протестировать функцию, которая определяет, совпадают ли в заданной строке первая и последняя буквы.

Задача 52

Написать и протестировать функцию, переставляющую в прямоугольной матрице строки в обратном порядке.

Задача 53

Даны натуральные числа n и m . Написать и протестировать функцию, которая возвращает результат операции сложения двух чисел, образованных k младшими цифрами числа n и k старшими цифрами числа m .

Задача 54

Написать и протестировать функцию, которая преобразует строку 16-х цифр в эквивалентное ей целое 10-е число.

Задача 55

Написать функцию для вычисления значения

$$S(n) = \sum_{i=1}^n \frac{(2i)!}{(n+i)!}.$$

Вычислить с ее помощью значения $S(n)$ для n от 12 до 24 с шагом 4.

Задача 56

Дана прямоугольная вещественная матрица. Проверить, упорядочены ли по неубыванию суммы элементов строк этой матрицы.

Задача 57

Написать и протестировать функцию для определения полярных координат точки по ее прямоугольным декартовым координатам. Зависимость полярных и декартовых координат:

$$r = \sqrt{x^2 + y^2}; \quad \varphi = \arctg(y / x).$$

Задача 58

Даны натуральные числа n , m и вещественные массивы $a[n]$, $b[m]$, $c[30]$. Найти

$$l = \begin{cases} \min_i(b_i) + \min_i(c_i) & \text{при } \left| \min_i(a_i) \right| > 1, \\ 1 + \max_i^2(c_i). \end{cases}$$

Задача 59

Написать и протестировать функцию, которая определяет, входит ли каждая буква в заданную строку не менее двух раз.

Задача 60

Даны натуральные числа n , m и k . Написать и протестировать функцию, которая возвращает суммы, полученные в результате сложения k младших цифр числа n и k старших цифр числа m .

Задача 61↑

Написать и протестировать функцию перестановки строк матрицы согласно вектору транспозиции.

Задача 62

Написать и протестировать функцию, которая по заданной строке Str формирует новую строку, состоящую только из несовпадающих цифр, входящих в Str .