

# Программирование

## Лабораторная работа №

Карпов Р. В., ИВТ-2

### 1. Тема лабораторной работы

Математические операции. Переменные и их типы. Операторы. Циклы.  
Простые условные конструкции. Основы работы со статическими массивами.

### Задание 1.2

### 2. Постановка задачи

Написать простую программу. Ввести два числа с клавиатуры, вычислить их сумму и напечатать результат. Использовать функцию `printf` для приглашений на ввод и для распечатки результата. Использовать функцию `scanf` для ввода каждого числа отдельно с клавиатуры. Для получения доступа к функциям `printf` и `scanf` включить в программу заголовочный файл `stdio.h`. Использовать корректные спецификаторы форматирования. Здесь и далее для распечатки надписей на экране использовать латинские буквы для избежания проблем с кодировками символов.

### 3. Математическая модель

$$a + b$$

### 4. Список идентификаторов

Имя	Тип	Смысл
a	int	Число a
b	int	Число b
result	int	Результат складывания чисел a и b

## 5. Код программы

```
#include <stdio.h>

int main() {
    int a, b;

    printf("Vvedite chislo a: ");
    scanf("%d", &a);

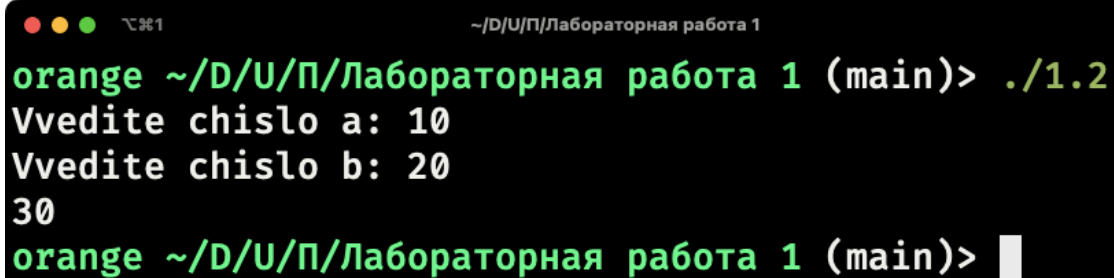
    printf("Vvedite chislo b: ");
    scanf("%d", &b);

    int result = a + b;

    printf("%d\n", result);

    return 0;
}
```

## 6. Результаты выполненной работы



```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./1.2
Vvedite chislo a: 10
Vvedite chislo b: 20
30
orange ~/D/U/П/Лабораторная работа 1 (main)> 
```

## Задание 1.3

### 2. Постановка задачи

Вычислить значение выражения

### 3. Математическая модель

$$u(x, y) = \frac{1 + \sin^2(x + y)}{2 + \left| x - \frac{2x^2}{1 + |\sin(x + y)|} \right|}$$

### 4. Список идентификаторов

Имя	Тип	Смысл
x	float	Число x
y	float	Число y
up	float	Числитель дроби
bottom	float	Знаменатель дроби
result	float	Результат деления up на bottom

### 5. Код программы

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

int main() {
    float x, y;

    printf("Vvedite x: ");
    scanf("%f", &x);

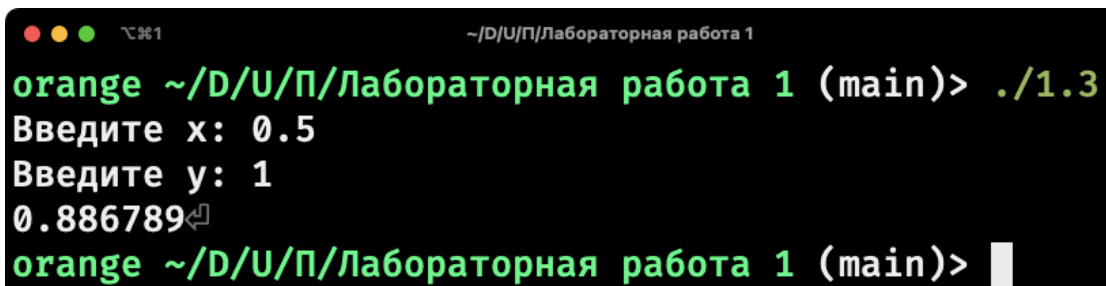
    printf("Vvedite y: ");
    scanf("%f", &y);

    float up = 1.0f + sinf(x + y) * sinf(x + y);
    float bottom = 2.0f + fabsf(x - (2.0f * x*x) / (1.0f +
fabsf(sinf(x+y))));
    float result = up / bottom;

    printf("%f", result);

    return 0;
}
```

## 6. Результаты выполненной работы



A terminal window titled "orange ~/D/U/П/Лабораторная работа 1" with standard macOS window controls. The terminal shows the execution of a program. The prompt is "orange ~/D/U/П/Лабораторная работа 1 (main)>". The user enters "./1.3", and the program prompts "Введите x: 0.5". The user enters "Введите y: 1", and the program outputs "0.886789" followed by a cursor. The prompt returns to "orange ~/D/U/П/Лабораторная работа 1 (main)>".

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./1.3
Введите x: 0.5
Введите y: 1
0.886789
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```

## Задание 1.4

### 2. Постановка задачи

Вычислить значение выражения. Выполнить для следующих значений:

$$a = 0.12, b = 3.5, c = 2.4, x = 1.4;$$

$$a = 0.12, b = 3.5, c = 2.4, x = 1.6;$$

$$a = 0.27, b = 3.9, c = 2.8, x = 1.8;$$

### 3. Математическая модель

$$h(x) = -\frac{x - a}{\sqrt[3]{x^2 + a^2}} - \frac{4\sqrt[4]{(x^2 + b^2)^3}}{2 + a + b + \sqrt[3]{(x - c)^2}}$$

### 4. Список идентификаторов

Имя	Тип	Смысл
a	float	Число a
b	float	Число b
c	float	Число c
x	float	Число x
q	float	Промежуточные вычисления
h	float	Результат вычислений

## 5. Код программы

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

int main() {
    float x, a, b, c;

    printf();
    scanf("%f", &x);

    printf();
    scanf("%f", &a);

    printf();
    scanf("%f", &b);

    printf();
    scanf("%f", &c);

    float q = x*x + b*b;

    float h = -(x - a) / powf(x*x + a*a, 1.0f / 3.0f) - (4.0f *
powf(q*q*q, 1.0f / 4.0f)) / (2.0f + a + b + powf((x-c)*(x-c), 1.0f /
3.0f));
    printf("%f", h);

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./1.4
Vvedite a: 0.12
Vvedite b: 3.5
Vvedite c: 2.4
Vvedite x: 1.4
-5.442602
orange ~/D/U/П/Лабораторная работа 1 (main)> ./1.4
Vvedite a: 0.12
Vvedite b: 3.5
Vvedite c: 2.4
Vvedite x: 1.6
-5.738755
orange ~/D/U/П/Лабораторная работа 1 (main)> ./1.4
Vvedite a: 0.27
Vvedite b: 3.9
Vvedite c: 2.8
Vvedite x: 1.8
-5.992693
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```

## Задание 2.1

### 2. Постановка задачи

Вычислить используя цикл for координаты планеты Марс относительно Земли с течением времени  $t$ . Распечатать на экране координаты для каждой итерации по  $t$ . Координаты планеты Марс для каждой итерации задаются заданы формулами:

$$x = r_1 \cos(w_1 t) - r_2 \cos(w_2 t),$$

$$y = r_1 \sin(w_1 t) - r_2 \sin(w_2 t),$$

$$w_1 = \frac{2\pi}{T_1},$$

$$w_2 = \frac{2\pi}{T_2},$$

где  $r_1$  – радиус орбиты Марса,  $r_2$  – радиус орбиты Земли,  $T_1$  и  $T_2$  — периоды обращения указанных планет соответственно,  $t$  – каждый заданный момент времени внутри цикла по времени. Подберите подходящие единицы измерения для времени и расстояния.

### 3. Математическая модель

$$x = r_1 \cos(w_1 t) - r_2 \cos(w_2 t),$$

$$y = r_1 \sin(w_1 t) - r_2 \sin(w_2 t),$$

$$w_1 = \frac{2\pi}{T_1},$$

$$w_2 = \frac{2\pi}{T_2},$$

### 4. Список идентификаторов

Имя	Тип	Смысл
mars_r	float	Радиус орбиты Марса
earth_r	float	Радиус орбиты Земли
mart_t	float	Период обращения Марса вокруг солнца
earth_t	float	Период обращения Земли вокруг солнца
mars_w	float	Омега для Марса
earth_w	float	Омега для Земли
t	float	Время

## 5. Код программы

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

int main() {
    float mars_r = 228.0f * 1000000.0f * 1000.0f;
    float earth_r = 149.0f * 1000000.0f * 1000.0f;

    float mars_t = 687.0f * 24.0f * 60.0f * 60.0f;
    float earth_t = 365.0f * 24.0f * 60.0f * 60.0f;

    float mars_w = 2.0f * M_PI / mars_t;
    float earth_w = 2.0f * M_PI / earth_t;

    for (float t = 0.0f; t < 1000.0f * 1000.0f * 1000.0f; t += 1000.0f) {
        float x = mars_r * cosf(mars_w * t) - earth_r * cosf(earth_w * t);
        float y = mars_r * sinf(mars_w * t) - earth_r * sinf(earth_w * t);

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

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./2.1 | tail
168162574336.000000, -42666459136.000000
168153939968.000000, -42645061632.000000
168145289216.000000, -42623664128.000000
168134426624.000000, -42601709568.000000
168125759488.000000, -42580328448.000000
168118534144.000000, -42557947904.000000
168107671552.000000, -42535993344.000000
168099020800.000000, -42514628608.000000
168088141824.000000, -42492674048.000000
168080900096.000000, -42470326272.000000
orange ~/D/U/П/Лабораторная работа 1 (main)> 
```



## Задание 2.2

### 2. Постановка задачи

Вычислить определённый интеграл от заданной функции методом трапеций:

$$\int_a^b f(x)dx = \int_a^b e^{x+2}dx$$

Функция  $f(x)$  может быть выбрана и самостоятельно. Результат интегрирования сравнить с вычисленным вручную и убедиться в корректности результата.

### 3. Математическая модель

$$\int_a^b f(x)dx = \int_a^b e^{x+2}dx$$

### 4. Список идентификаторов

Имя	Тип	Смысл
a	double	Нижний предел интеграла
b	double	Верхний предел интеграла
n	int	Количество шагов
h	double	Шаг
x	double	Точка на оси x
result	double	Результат вычисления интеграла

## 5. Код программы

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

static double f(double x) {
    return exp(x + 2.0);
}

int main() {
    double a, b;
    int n;

    printf("Vvedite a: ");
    scanf("%lf", &a);

    printf("Vvedite b: ");
    scanf("%lf", &b);

    printf("Vvedite n: ");
    scanf("%d", &n);

    double h = (b - a) / n;
    double result = (f(a) + f(b)) / 2.0;
    for (double x = a + h; x ≤ b - h; x += h) {
        result += f(x);
    }

    result *= h;

    printf("%lf\n", result);

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./2.2
Vvedite a: 0
Vvedite b: 1
Vvedite n: 100000
12.696481
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```

## Задание 2.3

### 2. Постановка задачи

Организовать и распечатать последовательность чисел Падована, не превосходящих число  $m$ , введенное с клавиатуры. Числа Падована представлены следующим рядом: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86, 114, 151, 200, 265, ... Использовать конструкцию `for` и простые варианты условной конструкции `if else`. Для этих чисел заданы формулы:

$$P(0) = P(1) = P(2) = 1,$$

$$P(n) = P(n - 2) + P(n - 3).$$

### 3. Математическая модель

$$P(0) = P(1) = P(2) = 1,$$

$$P(n) = P(n - 2) + P(n - 3)$$

### 4. Список идентификаторов

Имя	Тип	Смысл
<code>m</code>	<code>unsigned long long</code>	Количество чисел в последовательности
<code>first</code>	<code>unsigned long long</code>	Первое из предыдущих 4 чисел
<code>second</code>	<code>unsigned long long</code>	Второе из предыдущих 4 чисел
<code>third</code>	<code>unsigned long long</code>	Третье из предыдущих 4 чисел
<code>fourth</code>	<code>unsigned long long</code>	Четвёртое из предыдущих 4 чисел

## 5. Код программы

```
#include <stdio.h>

int main() {
    unsigned long long m;

    printf("Vvedite m: ");
    scanf("%llu", &m);

    unsigned long long first = 1;
    unsigned long long second = 1;
    unsigned long long third = 1;
    unsigned long long fourth = 2;

    for (unsigned long long n = 0; fourth ≤ m; ++n) {
        if (n ≤ 2) {
            printf("%llu\n", 1);
        } else {
            printf("%llu\n", fourth);
            fourth = first + second;
            first = second;
            second = third;
            third = fourth;
        }
    }

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./2.3
Vvedite m: 21
1
1
1
2
2
2
3
4
5
7
9
12
16
21
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```

## Задание 2.4

### 2. Постановка задачи

С клавиатуры вводится трёх значное число, считается сумма его цифр. Если сумма цифр числа больше 10, то вводится следующее трёхзначное число, если сумма меньше либо равна 10 — программа завершается.

### 3. Математическая модель

$$sum = \frac{n}{100} + \frac{n}{10} \bmod 10 + n \bmod 10$$

### 4. Список идентификаторов

Имя	Тип	Смысл
sum	int	Сумма цифр числа
num	int	Число

### 5. Код программы

```
#include <stdio.h>

int main() {
    int sum = 0;

    do {
        int num;
        printf("Vvedite chislo: ");
        scanf("%d", &num);

        sum = num / 100 + (num / 10) % 10 + num % 10;
        printf("Summa tsifr: %d\n", sum);
    } while (sum > 10);

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./2.4
Vvedite chislo: 123
Summa tsifr: 6
orange ~/D/U/П/Лабораторная работа 1 (main)> ./2.4
Vvedite chislo: 999
Summa tsifr: 27
Vvedite chislo: 444
Summa tsifr: 12
Vvedite chislo: 442
Summa tsifr: 10
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```

## Задание 3.1

### 2. Постановка задачи

Для некоторого числового вектора  $X$ , введённого с клавиатуры, вычислить значения вектора  $Y = X \cdot X$  ( $y_i = x_i \cdot x_i$  — поэлементно).

### 3. Математическая модель

$$y_i = x_i \cdot x_i$$

### 4. Список идентификаторов

Имя	Тип	Смысл
$X$	float[3]	Исходный вектор
$Y$	float[3]	Получившейся вектор

### 5. Код программы

```
#include <stdio.h>

int main() {
    float X[3];

    printf("Vvedite vector iz tryoh elementov: ");
    scanf("%f %f %f", &X[0], &X[1], &X[2]);

    float Y[3];

    for (int i = 0; i < 3; ++i) {
        Y[i] = X[i] * X[i];
    }

    printf("%f %f %f\n", Y[0], Y[1], Y[2]);

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./3.1
Vvedite vector iz tryoh elementov: 5 6 7
25.000000 36.000000 49.000000
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```



## Задание 3.2

### 2. Постановка задачи

Для некоторого числового массива  $X$ , введённого с клавиатуры поэлементно, изменить порядок элементов на обратный и распечатать результат на экране.

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
$X$	$\text{int}[10]$	Числовой вектор
left	int	Индекс элемента с начала
right	int	Индекс элемента с конца
temp	int	Переменная для сохранения значения элемента при обмене

## 5. Код программы

```
#include <stdio.h>

#define SIZE 10

int main() {
    int X[SIZE];

    for (int i = 0; i < SIZE; ++i) {
        printf("Vvedite %d element massiva: ", i + 1);
        scanf("%d", &X[i]);
    }

    int left = 0;
    int right = SIZE - 1;
    while (left < right) {
        int temp = X[left];
        X[left] = X[right];
        X[right] = temp;

        left++;
        right--;
    }

    printf("Massiv v obratnom poryadke:\n");
    for (int i = 0; i < 10; ++i) {
        printf("%d ", X[i]);
    }
    printf("\n");

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./3.2
Vvedite 1 element massiva: 1
Vvedite 2 element massiva: 2
Vvedite 3 element massiva: 3
Vvedite 4 element massiva: 4
Vvedite 5 element massiva: 5
Vvedite 6 element massiva: 6
Vvedite 7 element massiva: 7
Vvedite 8 element massiva: 8
Vvedite 9 element massiva: 9
Vvedite 10 element massiva: 10
Massiv v obratnom poryadke:
10 9 8 7 6 5 4 3 2 1
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```

## Задание 3.3

### 2. Постановка задачи

Транспонировать матрицу:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
A	int[3][3]	Матрица A

## 5. Код программы

```
#include <stdio.h>

int main() {
    int A[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    printf("\n");
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }

    for (int i = 0; i < 3; ++i) {
        for (int j = i; j < 3; ++j) {
            int temp = A[j][i];
            A[j][i] = A[i][j];
            A[i][j] = temp;
        }
    }

    printf("\n");
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./3.3
Матрица A:
1 2 3
4 5 6
7 8 9
Транспонированная матрица A:
1 4 7
2 5 8
3 6 9
orange ~/D/U/П/Лабораторная работа 1 (main)> 
```

## Задание 3.4

### 2. Постановка задачи

Преобразовать исходную матрицу так, чтобы первый элемент каждой строки был заменён средним арифметическим элементов этой строки.

### 3. Математическая модель

$$S = \frac{\sum_{i=1}^n x_i}{n}$$

### 4. Список идентификаторов

Имя	Тип	Смысл
A	int[4][4]	Матрица A
sum	int	Сумма элементов в строке

## 5. Код программы

```
#include <stdio.h>

int main() {
    int A[4][4];

    for (int i = 0; i < 4; ++i) {
        printf("Vvedite %d stroku matritsi (4 elementa): ", i + 1);
        scanf("%d %d %d %d", &A[i][0], &A[i][1], &A[i][2], &A[i][3]);
    }

    printf("Ishodnaya matritsa:\n");
    for (int i = 0; i < 4; ++i) {
        for (int j = 0; j < 4; ++j) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }

    for (int i = 0; i < 4; ++i) {
        int sum = 0;
        for (int j = 0; j < 4; ++j) {
            sum += A[i][j];
        }

        A[i][0] = sum / 4;
    }

    printf("Preobrazovannaya matritsa:\n");
    for (int i = 0; i < 4; ++i) {
        for (int j = 0; j < 4; ++j) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./3.4
Vvedite 1 stroku matritsi (4 elementa): 5 10 15 20
Vvedite 2 stroku matritsi (4 elementa): 25 30 35 40
Vvedite 3 stroku matritsi (4 elementa): 45 50 55 60
Vvedite 4 stroku matritsi (4 elementa): 65 70 75 80
Ishodnaya matritsa:
5 10 15 20
25 30 35 40
45 50 55 60
65 70 75 80
Preobrazovanaya matritsa:
12 10 15 20
32 30 35 40
52 50 55 60
72 70 75 80
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```



## Задание 3.5

### 2. Постановка задачи

Реализовать самостоятельно алгоритм сортировки вставками (без создания своих функций, внутри функции main). Пример: <https://habr.com/ru/post/181271/>

### 3. Математическая модель

Отсутствует

### 4. Список идентификаторов

Имя	Тип	Смысл
М	int[10]	Массив

## 5. Код программы

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

#define SIZE 10

int main() {
    srand(time(NULL));

    int M[SIZE];

    for (int i = 0; i < SIZE; ++i) {
        M[i] = rand() % 100;
    }

    printf("Ishodniy massiv:\n");
    for (int i = 0; i < SIZE; ++i) {
        printf("%d ", M[i]);
    }
    printf("\n");

    for (int i = 1; i < SIZE; ++i) {
        for (int j = i; j > 0; --j) {
            if (M[j - 1] > M[j]) {
                int temp = M[j - 1];
                M[j - 1] = M[j];
                M[j] = temp;
            } else {
                break;
            }
        }
    }

    printf("Otsortirovanniy massiv:\n");
    for (int i = 0; i < SIZE; ++i) {
        printf("%d ", M[i]);
    }
    printf("\n");

    return 0;
}
```

## 6. Результаты выполненной работы

```
orange ~/D/U/П/Лабораторная работа 1 (main)> ./3.5
Ishodniy massiv:
95 39 36 93 30 34 1 57 15 39
Otsortirovanniy massiv:
1 15 30 34 36 39 39 57 93 95
orange ~/D/U/П/Лабораторная работа 1 (main)> █
```