

# Лабораторная работа №1

## Математические операции.

### Переменные и их типы. Операторы. Циклы.

### Простые условные конструкции.

### Основы работы со статическими массивами.

#### Задание 1.2

##### 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.

##### Математическая модель:

$c = a + b$

##### Список идентификаторов:

Имя	Тип	Назначение
a	int	Первое слагаемое.
b	int	Второе слагаемое.
c	int	Сумма a и b.

##### Код программы:

```
1 #include <stdio.h>
2
3 int
4 main(){
5     int a, b, c;
6     printf("Enter 2 nubers one by one:\n");
7     scanf("%d", &a);
8     scanf("%d", &b);
9     c = a+b;
10    printf("%d\n", c);
11    return 0;}
```

Результаты выполненной работы:

```
Enter 2 nubers one by one:
10
14
24
```

### Задание 1.3

Постановка задачи:

Вычислить значение выражения:

Математическая модель:

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

Список идентификаторов:

Имя	Тип	Назначение
x	double	Вводимый пользователем первый аргумент функции.
y	double	Вводимый пользователем второй аргумент функции.
rez	double	Значение функции от x и y.

Код программы:

```
1 #include <stdio.h>
2 #include <math.h>
3
4
5 int
6 main(){
7     double x,y;
8     scanf("%lf %lf", &x, &y);
9
10    double rez = ( 1+sin(x+y)*sin(x+y) )/
11    ( 2 + fabs(x - (2*x*x)/(1+fabs(sin(x+y)))));
12
13    printf("%lf\n", rez);
14    return 0;}
```

Результаты выполненной работы:

```
1 10
0.999988
```

## Задание 1.4

Постановка задачи:

Вычислить значение выражения:

Математическая модель:

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

Список идентификаторов:

Имя	Тип	Назначение
x	double	Вводимый пользователем первый аргумент функции.
y	double	Вводимый пользователем второй аргумент функции.
rez	double	Значение функции от x и y.

Код программы:

```
1 #include <stdio.h>
2 #include <math.h>
3
4
5 int
6 main(){
7     double a, b, c, x, rez;
8     printf("Enter a b c x\n");
9     scanf("%lf %lf %lf %lf", &a, &b, &c, &x);
10    rez = -1*(x-a)/cbrtf(x*x + a*a)
11          -1*(4*pow((x*x + b*b), 3/4))/
12          (2+a+b*pow((x-c), 2/3));
13    printf("%lf\n", rez);
14    return 0;}
```

Результаты выполненной работы:

```
Enter a b c x
0.12 3.5 2.4 1.4
-1.624538
Enter a b c x
0.12 3.5 2.4 1.6
-1.684095
Enter a b c x
0.27 3.9 2.8 1.8
-1.584214
```

## Задание 2.1

### Постановка задачи:

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

### Математическая модель:

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

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

$$w = \frac{2\pi}{T}$$

### Список идентификаторов:

Имя	Тип	Назначение
x	double	Координата X.
y	double	Координата Y.
mR	double	Радиус Марса.
eR	double	Радиус Земли.
mT	double	Период Марса.
eT	double	Период земли
mW	double	Коэффициент $t$ Марса.
eW	double	Коэффициент $t$ Земли.
t	unsigned long	Время.

### Код программы:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 #define PI 3.14159
5
6 int
7 main(){
8     double x, y;
9     double mR = 240310000, eR = 149900000,
10         mT = 687, eT = 365.24;
11
12     double mW = 2*PI/mT, eW = 2*PI/eT;
13     for(unsigned long t= 1; t <= pow(10, 4); t+= 1000 ){
14         x = mR * cos(mW*t) - eR * cos(eW*t);
15         y = mR * sin(mW*t) - eR * sin(eW*t);
16         printf("x:%lf\ty:%lf\n", x, y);
17     }
18     return 0;}
```

Результаты выполненной работы:

x:90422129.664628	y:-380782.623285
x:-222825427.772905	y:213696188.331797
x:353570161.224874	y:-145463005.274979
x:-193895057.626225	y:30368963.039578
x:-36226348.761132	y:-172554476.670489
x:8870492.974795	y:376376900.018642
x:113239151.462739	y:-302840648.104267
x:13820618.538838	y:93273887.494597
x:-270329837.290737	y:-107681537.256883
x:285312727.174762	y:261431795.956480

## Задание 2.2

Постановка задачи:

Вычислить определённый интеграл от заданной функции методом трапеций.

Математическая модель:

$$\int_a^b e^{x+2} dx$$

Список идентификаторов:

Имя	Тип	Назначение
a	double	Начало отрезка интегрирования.
b	double	Конец отрезка интегрирования.
c	double	Количество значений функции, которые найдёт программа.
s	double	Сумма найденных на заданном отрезке значений функции.
h	double	Размер одного «шага» по заданному диапазону.
x	double	Аргумент заданной функции.

Код программы:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int
5 main(){
6     double a, b, c, s = 0, h;
7     printf("Enter from to and amount of steps:\n");
8
9     scanf("%lf %lf %lf", &a, &b, &c);
10    h = (b-a)/c;
11    for(double x = a+h; x <= b-h; x += h){
12        s += exp(x+2);
13    }
14    s = h * ( (exp(a+2) + exp(b+2))/2 + s);
15    printf("result:%lf\n", s);
16    return 0;}
```

Результаты выполненной работы:

```
Enter from to and amount of steps:
1 2 150
result:34.512741
```

## Задание 2.3

Постановка задачи:

Организовать и распечатать последовательность чисел Падована, не превосходящих число  $m$ , введённое 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`.

Математическая модель:

$$P(0)=P(1)=P(2)=1$$
$$P(n)=P(n-2)+P(n-3)$$

Список идентификаторов:

Имя	Тип	Назначение
m	int	Количество чисел которые необходимо вывести
p	unsigned int [128]	Массив чисел
n	int	Аргумент цикла.

Код программы:

```
1 #include <stdio.h>
2
3 int
4 main(){
5     int m;
6     scanf("%d", &m);
7
8     unsigned int p[128] = {1,1,1};
9     for(int n = 3; n <= m; n++){
10         p[n] = p[n-2] + p[n-3];
11     }
12     for(int n = 0; n <= m; n++){
13         printf("%d, ", p[n]);
14     }
15     putchar('\n');
16     return 0;}
```

Результаты выполненной работы:

```
21
1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86, 114, 151, 200, 265
```

## Задание 2.4

Постановка задачи:

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

Математическая модель:

$$\begin{aligned}c &= \text{mod}(n, 10) \\ b &= (\text{mod}(n, 100) - c) / 10 \\ a &= (n - b - c) / 100\end{aligned}$$

Список идентификаторов:

Имя	Тип	Назначение
n	int	Введённое число.
a	int	Количество сотен.
b	int	Количество десятков.
c	int	Количество единиц.

Код программы:

```
1 #include <stdio.h>
2
3 int
4 main()
5 {
6     int n, a, b, c;
7     while (1){
8         scanf("%d", &n);
9         c = n%10;
10        b = (n%100 - c)/10;
11        a = (n - b - c)/100;
12        if (a+b+c<=10)
13            break;
14    }
15    return 0;}
```

Результаты выполненной работы:

```
>while true; do ./a.out ; echo ret:$?; done
999
129
541
ret:0
540
ret:0
```



## Задание 3.1

### 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).

### Математическая модель:

$$y_i = x_i * x_i$$

### Список идентификаторов:

Имя	Тип	Назначение
n	int	Размер числового вектора.
x	int[128]	Введённый числовой вектор.
y	int[128]	Результат поэлементно произведения $x$ на $x$ .
i	int	Аргумент циклов.

### Код программы:

```
1 #include <stdio.h>
2
3 int
4 main() {
5     int n, X[128], Y[128];
6
7     printf("Enter the size of the vector: \n");
8     scanf("%d", &n);
9
10    printf("Enter the elements of the vector X:\n");
11    for (int i = 0; i < n; i++) {
12        scanf("%d", &X[i]);
13    }
14
15    for (int i = 0; i < n; i++) {
16        Y[i] = X[i] * X[i];
17    }
18
19    for (int i = 0; i < n; i++) {
20        printf("%d ", Y[i]);
21    }
22    putchar('\n');
23
24    return 0;}
```

Результаты выполненной работы:

```
Enter the size of the vector:
5
Enter the elements of the vector X:
2
3
4
5
6
4 9 16 25 36
```

### Задание 3.2

Постановка задачи:

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

Математическая модель:

-

Список идентификаторов:

Имя	Тип	Назначение
len	int	Размер массива.
temp	int	Переменная для замены элементов массива.
arr	int[128]	Введённый массив.
i	int	Аргумент циклов.

Код программы:

```
1 #include <stdio.h>
2
3 int
4 main(){
5     int len, temp;
6     scanf("%d", &len);
7     len = len - 1;
8     int arr[128] = {0};
9
10    for(int i = 0; i <= len ; i++)
11        scanf("%d", &arr[i]);
12
13    for(int i = 0; i <= len/2; i++){
14        temp = arr[i];
15        arr[i] = arr[len-i];
16        arr[len-i] = temp;
17    }
18
19    for(int i = 0; i <= len; i++)
20        printf("%d ", arr[i]);
21
22    putchar('\n');
23
24    return 0;}
```

Результаты выполненной работы:

```
10
1
2
3
4
5
6
7
8
9
10
10 9 8 7 6 5 4 3 2 1
```

### Задание 3.3

Постановка задачи:

Транспонировать матрицу:

Математическая модель:

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

Список идентификаторов:

Имя	Тип	Назначение
temp	int	Переменная для замены элементов матрицы.
a	int[3][3]	Матрица 3 на 3.
i	int	Аргумент циклов.
j	int	Аргумент циклов.

Код программы:

```
1 #include <stdio.h>
2
3 int
4 main(){
5     int temp, a[3][3] = {
6         {1,2,3},
7         {4,5,6},
8         {7,8,9}
9     };
10    for(int i = 0; i <= 2; i++){
11        for(int j = 0; j <= 2; j++){
12            if(i <= j){
13                temp = a[i][j];
14                a[i][j] = a[j][i];
15                a[j][i] = temp;
16            }
17        }
18    } for(int i = 0; i <= 2; i++){
19        for(int j = 0; j <= 2; j++){
20            printf("%d ", a[i][j]);
21        }
22        putchar('\n');
23    }
24
25    return 0;}
```

Результаты выполненной работы:

1	4	7
2	5	8
3	6	9

### Задание 3.4

Постановка задачи:

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

Математическая модель:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \rightarrow \begin{pmatrix} \frac{(a_{11}+a_{12}+a_{13})}{3} & a_{12} & a_{13} \\ \frac{(a_{21}+a_{22}+a_{23})}{3} & a_{22} & a_{23} \\ \frac{(a_{31}+a_{32}+a_{33})}{3} & a_{32} & a_{33} \end{pmatrix}$$

Список идентификаторов:

Имя	Тип	Назначение
temp	int	Переменная для нахождения среднего арифметического.
a	int[3][3]	Матрица 3 на 3.
i	int	Аргумент циклов.
j	int	Аргумент циклов.

Код программы:

```
1 #include <stdio.h>
2
3 int
4 main(){
5     float a[3][3] = {
6         {1,2,3},
7         {4,5,6},
8         {7,8,9}
9     };
10    float temp = 0;
11
12    for(int i = 0; i <= 2; i++){
13        for(int j = 0; j <= 2; j++){
14            temp += a[i][j];
15        }
16        a[i][0] = temp/3;
17        temp = 0;
18    } for(int i = 0; i <= 2; i++){
19        for(int j = 0; j <= 2; j++){
20            printf("%.3f ", a[i][j]);
21        }
22        putchar('\n');
23    }
24    return 0;}
```

Результаты выполненной работы:

```
2.000 2.000 3.000
5.000 5.000 6.000
8.000 8.000 9.000
```

### Задание 3.5

Постановка задачи:

Реализовать самостоятельно алгоритм сортировки вставками (без создания своих функций, внутри функции main).

Математическая модель:

-

Список идентификаторов:

Имя	Тип	Назначение
len	int	Размер массива.
temp	int	Переменная для замены элементов массива.
arr	int[128]	Введённый массив.
i	int	Аргумент циклов.
j	int	Аргумент циклов.

Код программы:

```
1 #include <stdio.h>
2
3 int
4 main(){
5     int len, temp;
6     scanf("%d", &len);
7     len = len - 1;
8     int arr[128] = {0};
9
10    for(int i = 0; i <= len ; i++){
11        scanf("%d", &arr[i]);
12
13
14        for(int i = 0; i <= len; i++){
15            for (int j = i; j > 0; j-- ){
16                if (arr[j-1] > arr[j]){
17                    temp = arr[j-1];
18                    arr[j-1] = arr[j];
19                    arr[j] = temp;
20                }
21            }
22        }
23
24        for(int n = 0; n <= len; n++){
25            printf("%d ", arr[n]);
26
27            putchar('\n');
28        return 0;}
```

Результаты выполненной работы:

```
10
10
8
9
2
3
1
2
4
5
6
1 2 2 3 4 5 6 8 9 10
```