

Лабораторная работа №1. Задачи.

Комплект 1: Начала программирование.

Операторы, вычисления, ввод-вывод.

Задание 1.2. Написать простую программу.

Математическая модель: $c = x + y$

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
x	int	Вводимое число с клавиатуры
y	int	Вводимое число с клавиатуры
c	int	Результат вычисления

Код программы:

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

```
int main() {  
    int x, y, c;  
    printf("введите x:");  
    scanf("%d", &x);  
    printf("введите y:");  
    scanf("%d", &y);  
    c = x + y;  
    printf("сумма c = %d", c);  
  
    return 0;  
}
```

Результат выполненной работы:

```
введите x:2  
введите y:3  
сумма c = 5
```

Задание 1.3. 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	СМЫСЛ
x	int	Вводимое число с клавиатуры
y	int	Вводимое число с клавиатуры
u	double	Результат вычисления

Код программы:

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

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

```
int main() {
```

```
    double x, y, u;
```

```
    printf("Введите значение x: ");
```

```
    scanf("%lf", &x);
```

```
    printf("Введите значение y: ");
```

```
    scanf("%lf", &y);
```

```
    u = ((1 + (sin(x + y) * sin(x + y))) / (2 + (fabs(x - (2 * x * x))) / (1 + fabs(sin(x + y))))));
```

```
    printf("Значение выражения u(x, y) = %lf\n", u);
```

```
    return 0;
```

```
}
```

Результат выполненной работы:

```
Введите значение x: 1.2
Введите значение y: 3.7
Значение выражения u(x, y) = 0.690170
```

1.4. 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	СМЫСЛ
x	double	Переменная
a	double	Переменная
b	double	Переменная
c	double	Переменная
h	double	Результат вычислений

Код программы(1):

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

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

```
int main() {
```

```
    double h;
```

```
    double a = 0.12;
```

```
    double b = 3.5;
```

```
    double c = 2.4;
```

```
    double x = 1.4;
```

```
    h = -((x - a) / (pow(x * x + a * a, 1.0/3.0))) - (4 * (pow(pow(x * x + b * b, 3),  
1.0/4.0))) / (2 + a + b + (pow(pow(x - c, 2), 1.0/3.0)));
```

```
    printf("результат h = %f\n", h);
```

```
    return 0;
```

```
}
```

Код программы(2):

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

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

```
int main() {
```

```
    double h;
```

```
    double a = 0.12;
```

```
    double b = 3.5;
```

```
    double c = 2.4;
```

```
    double x = 1.6;
```

```
    h = -((x - a) / (pow(x * x + a * a, 1.0/3.0))) - (4 * (pow(pow(x * x + b * b, 3),  
1.0/4.0))) / (2 + a + b + (pow(pow(x - c, 2), 1.0/3.0)));
```

```
    printf("результат h = %f\n", h);
```

```
    return 0;
```

```
}
```

Код программы(3):

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

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

```
int main() {
```

```
    double h;
```

```
    double a = 0.27;
```

```
    double b = 3.9;
```

```
    double c = 2.8;
```

```
    double x = 1.8;
```

```
    h = -((x - a) / (pow(x * x + a * a, 1.0/3.0))) - (4 * (pow(pow(x * x + b * b, 3),  
1.0/4.0))) / (2 + a + b + (pow(pow(x - c, 2), 1.0/3.0)));
```

```
printf("результат h = %f\n", h);
```

```
return 0;
```

```
}
```

Результат выполненной работы(1):

```
результат h = -5.442602
```

Результат выполненной работы(2):

```
результат h = -5.738755
```

Результат выполненной работы(3):

```
результат h = -5.992693
```

Комплект 2: Организация циклов. Условные конструкции.

Задание 2.1. Вычислить, используя цикл for координаты планеты Марс относительно Земли с течением времени t. Распечатать на экране координаты для каждой итерации по t.

Математическая модель:

$$x = r_1 \cos(\omega_1 t) - r_2 \cos(\omega_2 t),$$

$$y = r_1 \sin(\omega_1 t) - r_2 \sin(\omega_2 t),$$

$$\omega_1 = \frac{2\pi}{T_1},$$

$$\omega_2 = \frac{2\pi}{T_2},$$

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.

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
x	double	Результат вычисления
y	double	Результат вычисления
r1	double	Радиус орбиты Марса
r2	double	Радиус орбиты Земли

w1	double	Угловая скорость Марса
w2	double	Угловая скорость Земли
t	double	Временной промежуток
T1	double	Период обращения Марса
T2	double	Период обращения Земли

Код программы:

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

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

```
int main() {
```

```
    double r1 = 1.52;
```

```
    double r2 = 1;
```

```
    double T1 = 687;
```

```
    double T2 = 365;
```

```
    double w1 = 2 * 3.14 / T1;
```

```
    double w2 = 2 * 3.14 / T2;
```

```
    double t_start = 0;
```

```
    double t_end = 2 * T1;
```

```
    double t_step = 30;
```

```
    for (double t = t_start; t <= t_end; t += t_step) {
```

```
        double x = r1 * cos(w1 * t) - r2 * cos(w2 * t);
```

```
        double y = r1 * sin(w1 * t) - r2 * sin(w2 * t);
```

```
        printf("t = %lf, x = %lf, y = %lf\n", t, x, y);
```

```
}
```

```
return 0;
```

```
}
```

Результат выполненной работы:

```
t = 1080.000000, x = -2.334587, y = -0.393559
t = 1110.000000, x = -2.110067, y = -1.250463
t = 1140.000000, x = -1.547732, y = -1.968205
t = 1170.000000, x = -0.735635, y = -2.410202
t = 1200.000000, x = 0.184567, y = -2.494001
t = 1230.000000, x = 1.049295, y = -2.210318
t = 1260.000000, x = 1.709975, y = -1.624517
t = 1290.000000, x = 2.066018, y = -0.860223
t = 1320.000000, x = 2.085679, y = -0.069272
t = 1350.000000, x = 1.809771, y = 0.604312
```

Задание 2.2. Вычислить определенный интеграл от заданной функции методом трапеций

Математическая модель:

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

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
n	int	Количество подынтервалов
a	double	Переменная
b	double	Переменная
sum	double	Сумма
h	double	Результат вычислений
i	int	Шаг

Код программы:

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

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

```
int main() {
```

```
    double a = 0;
```

```
    double b = 1;
```

```
    int n = 1000;
```

```

double h = (b - a) / n;

double sum = 0.5 * (exp(a * a) + exp(b * b));

for (int i = 1; i < n; i++) {
    double x = a + i * h;
    sum += exp(x+2);
}

double result = sum * h;

printf("f(x) = e^(x^2) на интервале [%f, %f] = %f\n", a, b, result);

return 0;
}

```

Результат выполненной работы:

$f(x) = e^{(x^2)}$ на интервале $[0.000000, 1.000000] = 12.684604$

Задание 2.3. Организовать и распечатать последовательность чисел Падована, не превосходящих число m , введенное с клавиатуры.

Математическая модель:

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

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

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
P	int	Массив
n	int	Шаг
m	int	Вводимое число с клавиатуры

Код программы:

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



```
int main() {  
    int m;  
    printf("Введите число m: ");  
    scanf("%d", &m);  
  
    if (m <= 0) {  
        printf("Ошибка.\n");  
        return 1;  
    }  
    int P[100];  
    P[0] = 1;  
    P[1] = 1;  
    P[2] = 1;  
    printf("%d\n", P[0]);  
    printf("%d\n", P[1]);  
    printf("%d\n", P[2]);  
  
    for (int n = 3; ; n++) {  
        P[n] = P[n - 2] + P[n - 3];  
  
        if (P[n] > m) {  
            break;  
        }  
  
        printf("%d\n", P[n]);  
    }  
  
    return 0;  
}
```

Результат выполненной работы:

```
Введите число n: 5
1
1
1
2
2
3
4
5
```

Задание 2.4. 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:

$$(n / 100) + ((n / 10) \% 10) + (n \% 10)$$

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
sum	int	Сумма цифр числа
n	int	Вводимое число с клавиатуры

Код программы:

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

```
int main() {
```

```
    int n;
```

```
    while (1) {
```

```
        printf("Введите трёхзначное число: ");
```

```
        scanf("%d", &n);
```

```
        if (n < 100 || n > 999) {
```

```
            printf("Ошибка.\n");
```

```

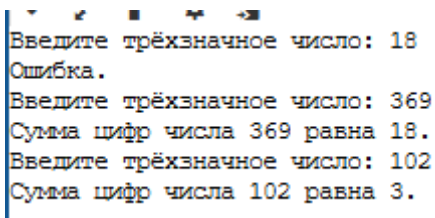
        continue;
    }

    int sum = (n / 100) + ((n / 10) % 10) + (n % 10);
    if (sum > 10) {
        printf("Сумма цифр числа %d равна %d.\n", n, sum);
    } else {
        printf("Сумма цифр числа %d равна %d.\n", n, sum);
        break;
    }
}

return 0;
}

```

Результат выполненной работы:



```

Введите трёхзначное число: 18
Ошибка.
Введите трёхзначное число: 369
Сумма цифр числа 369 равна 18.
Введите трёхзначное число: 102
Сумма цифр числа 102 равна 3.

```

Комплект 3: Основы работы со статическими массивами.

Задание 3.1: Для некоторого числового вектора X , введённого с клавиатуры, вычислить значения вектора $Y=XX(y_i=x_i * x_i$ —поэлементно).

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
X	double	Массив
Y	double	Массив
n	int	Размер массива
i	int	Параметр цикла

Код программы:

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

```

int main() {
    int n;

    printf("размер вектора X: ");
    scanf("%d", &n);

    double X[n], Y[n];

    printf("элементы вектора X:\n");
    for (int i = 0; i < n; i++) {
        scanf("%lf", &X[i]);
    }

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

    printf("элементы вектора Y:\n");
    for (int i = 0; i < n; i++) {
        printf("Y[%d] = %lf\n", i, Y[i]);
    }

    return 0;
}

```

Результат выполненной работы:

```

размер вектора X: 2
элементы вектора X:
5
4
элементы вектора Y:
Y[0] = 25.000000
Y[1] = 16.000000

```

Задание 3.2: Для некоторого числового массива X, введенного с клавиатуры поэлементно, изменить порядок элементов на обратный и распечатать результат на экране.

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
X	double	Массив
Y	double	Массив
n	int	Размер массива
i	int	Параметр цикла

Код программы:

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

```
int main() {
```

```
    int n;
```

```
    printf("размер массива X: ");
```

```
    scanf("%d", &n);
```

```
    double X[n];
```

```
    printf("элементы массива X:\n");
```

```
    for (int i = 0; i < n; i++) {
```

```
        scanf("%lf", &X[i]);
```

```
    }
```

```
    printf("элементы массива X в обратном порядке:\n");
```

```
    for (int i = n - 1; i >= 0; i--) {
```

```
        printf("%lf\n", X[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

Результат выполненной работы:

```
Введите размер массива X: 5
Введите элементы массива X:
4
5
6
9
2
Элементы массива X в обратном порядке:
2.000000
9.000000
6.000000
5.000000
4.000000
```

Задание 3.3: Транспонировать матрицу:

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

Математическая модель:

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
A_1	int	Транспонированная матрица
A	int	Матрица
i	int	Строки матрицы A
j	int	Столбцы матрицы A

Код программы:

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

```
int main() {
    int A[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
    int A_1;
```

```

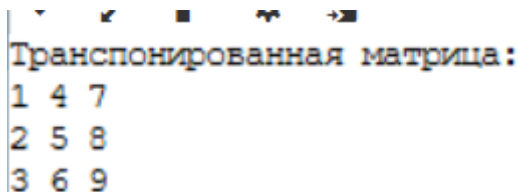
for (int i = 0; i < 3; i++) {
    for (int j = i + 1; j < 3; j++) {
        A_1 = A[i][j];
        A[i][j] = A[j][i];
        A[j][i] = A_1;
    }
}

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

return 0;
}

```

Результат выполненной работы:



```

Транспонированная матрица:
1 4 7
2 5 8
3 6 9

```

Задание 3.4. :Преобразовать исходную матрицу так, чтобы первый элемент каждой строки был заменён средним арифметическим элементов этой строки.

Математическая модель:

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
t	int	Переменная
A	int	Матрица
i	int	Строки матрицы A
j	int	Столбцы матрицы A
sum	int	Сумма

avg	double	Среднее арифметическое
-----	--------	---------------------------

Код программы:

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

```
int main() {  
    int A[3][3] = {  
        {1, 2, 3},  
        {4, 5, 6},  
        {7, 8, 9}  
    };  
  
    for (int i = 0; i < 3; i++) {  
        double sum = 0;  
        for (int j = 0; j < 3; j++) {  
            sum += A[i][j];  
        }  
        double avg = sum / 3;  
        A[i][0] = (int)avg;  
    }  
    printf("Преобразованная матрица:\n");  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 3; j++) {  
            printf("%d ", A[i][j]);  
        }  
        printf("\n");  
    }  
}
```



```
    return 0;
}
```

Результат выполненной работы:

Преобразованная матрица:

```
2 2 3
5 5 6
8 8 9
```

Задание 3.5 Реализовать самостоятельно алгоритм сортировки вставками (без создания своих функций, внутри функции main).

Математическая модель:

Список идентификаторов:

ИМЯ	ТИП	СМЫСЛ
n	int	Количество элементов в массиве
arr	int	Массив
k	int	Переменная
j	int	Позиция для вставки элемента
i	int	Перебор элементов в массиве

Код программы:

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

```
int main() {
```

```
    int arr[] = {12, 11, 10, 17, 9};
```

```
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    for (int i = 1; i < n; i++) {
```

```
        int k = arr[i];
```

```
        int j = i - 1;
```

```
        while (j >= 0 && arr[j] > k) {
```

```
    arr[j + 1] = arr[j];  
    j = j - 1;  
}  
arr[j + 1] = k;  
}
```

```
printf("отсортированный массив: \n");  
for (int i = 0; i < n; i++) {  
    printf("%d ", arr[i]);  
}  
printf("\n");
```

```
return 0;  
}
```

Результат выполненной работы:

```
отсортированный массив:  
9 10 11 12 17
```