

Лабораторная работа №1.

Математические операции. Переменные и их типы. Операторы. Циклы.

Простые условные конструкции. Основы работы со статическими массивами.

Комплект 1: Начала программирование.

1.1 Установить предложенную для работы среду программирования.

Был установлен Visual Studio Code, а также компилятор GCC

1.2 Написать простую программу. Ввести два числа с клавиатуры, вычислить их сумму и напечатать результат.

Программа:

```
C test.c > ...
1  #include <stdio.h>
2
3  int main(){
4      int s1, s2, c;
5      printf("Enter the number:");
6      scanf("%d", &s1);
7      printf("Enter the number:");
8      scanf("%d", &s2);
9      c = s1 + s2;
10     printf("%d", c);
11 }
```

Результат:

```
Enter the number:123
Enter the number:543
666
```

Список идентификаторов

Имя переменной	Тип	Смысл
s1	int	Первое вводимое число
s2	int	Второе вводимое число
c	int	Сумма чисел

1.3 Вычислить значение выражения.

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

введя x и y с клавиатуры. Подберите значения аргументов x и y самостоятельно за исключением тривиальных значений. Напечатайте вычисленное значение u(x, y) на экране. Включить в программу заголовочный файл math.h для доступа к математическим функциям.

Код:

```

1  #include <stdio.h>
2  #include <math.h>
3
4  double f(double x, double y) {
5      double n = 1 + pow(sin(x + y), 2);
6      double d = 2 + fabs(x - ((x * x) / (1 + fabs(sin(x + y)))));
7      return n / d;
8  }
9
10 int main() {
11     double x, y;
12     printf("Enter x and y: ");
13     scanf("%lf %lf", &x, &y);
14     printf("u(%.2f, %.2f) = %.5f\n", x, y, f(x, y));
15     return 0;
16 }
```

Результат:

```

Enter x and y: 52 34
u(52.00, 34.00) = 0.00137
```

Список идентификаторов:

Имя переменной	Тип	Смысл
f	double int	Функция, вычисляющая значение выражения
x	double	Переменная
y	double	Переменная
n	double	Временная переменная
d	double	Временная переменная

1.4

Вычислить значение выражения:

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

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.

```
1  #include <stdio.h>
2  #include <math.h>
3
4  double g(double x, double a, double b, double c) {
5      double t1 = (x - a) / pow((x * x + a * a), 1.0/3.0);
6      double t2 = (4.0 * pow(pow(x * x + b * b, 3), 1.0/4.0)) / ((2 + a + b + pow(pow((x - c),2), 1.0/3.0)));
7      return - t1 - t2;
8  }
9
10 int main() {
11     double a1 = 0.12, b1 = 3.5, c1 = 2.4, x1 = 1.4;
12     double a2 = 0.12, b2 = 3.5, c2 = 2.4, x2 = 1.6;
13     double a3 = 0.27, b3 = 3.9, c3 = 2.8, x3 = 1.2;
14
15     double r1 = g(x1, a1, b1, c1);
16     double r2 = g(x2, a2, b2, c2);
17     double r3 = g(x3, a3, b3, c3);
18
19     printf("h(%.2f) = %.5f (a=%.2f, b=%.2f, c=%.2f)\n", x1, r1, a1, b1, c1);
20     printf("h(%.2f) = %.5f (a=%.2f, b=%.2f, c=%.2f)\n", x2, r2, a2, b2, c2);
21     printf("h(%.2f) = %.5f (a=%.2f, b=%.2f, c=%.2f)\n", x3, r3, a3, b3, c3);
22
23     return 0;
24 }
```

Результат:

```
h(1.40) = -5.44260 (a=0.12, b=3.50, c=2.40)
h(1.60) = -5.73876 (a=0.12, b=3.50, c=2.40)
h(1.20) = -5.18398 (a=0.27, b=3.90, c=2.80)
```

Список идентификаторов:

Имя переменной	Тип	Смысл
g	double (double, double, double, double)	Функция

x	double	Входной аргумент для h(x)
r1, r2, r3	double	Результаты
t1, t2	double	Первая и вторая часть выражения
a1, b1, c1, x1	double	Входные аргументы

Комплект 2: Организация циклов. Условные конструкции.

2.1

Математическая модель:

$$\begin{aligned}
 x &= r_1 \cos(w_1 t) - r_2 \cos(w_2 t), \\
 y &= r_1 \sin(w_1 t) - r_2 \sin(w_2 t), \\
 w_1 &= \frac{2\pi}{T_1}, \\
 w_2 &= \frac{2\pi}{T_2},
 \end{aligned}$$

Код:

```

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

int main(){
    double PI = 3.14;
    double r1 = 227.94;
    double r2 = 149.6;
    double T1 = 687;
    double T2 = 365;
    double w1 = 2 * PI/T1;
    double w2 = 2 * PI/T2;
    double start = 0.0;
    double end = 10.0;
    printf("%5s %15s %15s\n", "Day", "x (million km)", "y (million km)");

    for (double t = start; t <= end; t +=1) {
        double x = r1 * cos(w1 * t) - r2 * cos(w2 * t);
        double y = r1 * sin(w1 * t) - r2 * sin(w2 * t);

        printf("%5.1f %15.8f %15.8f\n", t, x, y);
    }

    return 0;
}

```

Результат:

Day	x (million km)	y (million km)
0.0	78.34000000	0.00000000
1.0	78.35261896	-0.49019809
2.0	78.39047008	-0.97980838
3.0	78.45353608	-1.46824329
4.0	78.54178817	-1.95491566
5.0	78.65518609	-2.43923896
6.0	78.79367804	-2.92062751
7.0	78.95720079	-3.39849668
8.0	79.14567964	-3.87226308
9.0	79.35902847	-4.34134484
10.0	79.59714976	-4.80516173

Список идентификаторов:

Имя переменной	Тип	Смысл
PI	double	Число пи
T1, T2	double	Период обращения Марса вокруг Солнца и Земли вокруг Солнца (в днях)
r1, r2	double	Радиус орбиты Марса и Земли в миллионах километров
w1, w2	double	Угловая скорость Марса и Земли (рад/день)
start	double	Начальное значение времени
end	double	Конечное значение времени
x, y	double	Координата Марса относительно Земли по оси X и по оси Y(в млн км)

2.2 Вычислить определённый интеграл от заданной функции методом трапеций:

Математическая модель:

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

Код:

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

double f(double x) {
    return exp(x + 2);
}

double t(double a, double b, int n) {
    double h = (b - a) / n;
    double s = 0.5 * (f(a) + f(b));

    for (int i = 1; i < n; i++) {
        double x = a + i * h;
        s += f(x);
    }

    return s * h;
}

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

    printf("Enter a: ");
    scanf("%lf", &a);
    printf("Enter b: ");
    scanf("%lf", &b);
    printf("Enter n: ");
    scanf("%d", &n);

    double r = t(a, b, n);

    printf("Integral: %.6f\n", r);

    return 0;
}
```

Результат:

```
Enter a: 0
Enter b: 1
Enter n: 10.0
Integral: 12.707059
```

Список идентификаторов:

Имя переменной	Тип	Смысл
----------------	-----	-------

a	double	Нижний предел интегрирования
b	double	Верхний предел интегрирования
n	int	Количество разбиений
h	double	Шаг разбиения
s	double	Сумма значений функции в узлах трапеций
x	double	Текущая точка разбиения в методе трапеций
r	double	Итоговое значение приближенного интеграла
f(x)	double	Функция $f(x)=e^{x+2}$
t(a, b, n)	double	Функция, реализующая метод трапеций

2.3

Организовать и распечатать последовательность чисел Падована¹, 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`.

Математическая модель:

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

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

Код:

```

#include <stdio.h>

int main() {
    int m;
    printf("Enter m: ");
    scanf("%d", &m);

    if (m < 1) {
        printf("the padovan sequence starts with 1.\n");
        return 0;
    }

    int P0 = 1, P1 = 1, P2 = 1;
    printf("%d %d %d ", P0, P1, P2);

    for (int i = 3; ; i++) {
        int Pn = P0 + P1;

        if (Pn > m) {
            break;
        }

        printf("%d ", Pn);

        P0 = P1;
        P1 = P2;
        P2 = Pn;
    }

    printf("\n");
    return 0;
}

```

Результат:

```

Enter m: 13000
1 1 1 2 2 3 4 5 7 9 12 16 21 28 37 49 65 86 114 151 200 265 351 465 616 816 1081 1432 1897 2513 3329 4410 5842 7739 10252

```

Список идентификаторов:

Имя переменной	Тип	Смысл
a	double	Нижний предел интегрирования
b	double	Верхний предел интегрирования
n	int	Количество разбиений
h	double	Шаг разбиения

s	double	Сумма значений функции в узлах трапеций
x	double	Текущая точка разбиения в методе трапеций
r	double	Итоговое значение приближенного интеграла
f(x)	double	Функция $f(x)=e^{x+2}$
t(a, b, n)	double	Функция, реализующая метод трапеций

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.

Код:

```
#include <stdio.h>

int sum(int a) {
    return a / 100 + (a / 10) % 10 + a % 10;
}

int main() {
    int a;

    while (1) {
        printf("Enter a three-digit number: ");
        scanf("%d", &a);

        if (a < 100 || a > 999) {
            printf("Error: Enter a three-digit number\n");
            continue;
        }

        printf("The sum of the digits of a number: %d\n", sum(a));

        if (sum(a) <= 10) {
            printf("The program is completed\n");
            break;
        }

        printf("The amount is greater than 10, enter a new number.\n");
    }

    return 0;
}
```

Результат:

```
Enter a three-digit number: 546
The sum of the digits of a number: 15
The amount is greater than 10, enter a new number.
Enter a three-digit number: 123
The sum of the digits of a number: 6
The program is completed
```

Список идентификаторов:

Имя	Тип	Смысл
a	int	Хранит введённое пользователем трёхзначное число
sum(int a)	int	Функция, вычисляющая сумму цифр числа

3.1

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

Код:

```
#include <stdio.h>
#define SIZE 5

int main() {
    int X[SIZE], Y[SIZE];

    printf("Enter %d elements array X: ", SIZE);
    for (int i = 0; i < SIZE; i++) {
        scanf("%d", &X[i]);
    }
    for (int i = 0; i < SIZE; i++) {
        Y[i] = X[i] * X[i];
    }
    printf("Array Y: ");
    for (int i = 0; i < SIZE; i++) {
        printf("%d ", Y[i]);
    }
    printf("\n");

    return 0;
}
```

Результат:

```
Enter 5 elements array X: 1
2
3
4
5
Array Y: 1 4 9 16 25
```

Список идентификаторов:

Имя	Тип	Смысл
x	int	Исходный массив
y	int	Результат, возведенный в квадрат
i	int	Итератор цикла
SIZE	#define	Константа

3.2

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

Код:

```
#include <stdio.h>
#define SIZE 5

int main() {
    int X[SIZE];

    printf("Enter %d elements array: ", SIZE);
    for (int i = 0; i < SIZE; i++) {
        scanf("%d", &X[i]);
    }

    printf("Reverse order: ");
    for (int i = SIZE - 1; i >= 0; i--) {
        printf("%d ", X[i]);
    }
    printf("\n");

    return 0;
}
```

Результат массива:

```
Enter 5 elements array: 1
65
3
2
5
Reverse order: 5 2 3 65 1
```

Список идентификаторов:

Имя	Тип	Смысл
x	int	Исходный массив
i	int	Итератор цикла
SIZE	#define	Константа

3.3

Транспонировать матрицу:

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

Код:

```
#include <stdio.h>

#define N 3

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

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

    printf("Transposed matrix:\n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%d ", T[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Результат матрицы:

```
Transposed matrix:
1 4 7
2 5 8
3 6 9
```

Список идентификаторов:

Имя	Тип	Смысл
A	int	Исходная матрица
T	int	Транспонированная матрица
i, j	int	Итераторы циклов
N	#define	Константа

3.4

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

Код:

```
#include <stdio.h>
#define N 3

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

    printf("Enter the elements of a 3x3 matrix: \n");
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            scanf("%d", &A[i][j]);
        }
    }

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

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

    return 0;
}
```

Результат кода:

```

Enter the elements of a 3x3 matrix:
1
2
3
4
5
6
7
8
9
Modified matrix:
2 2 3
5 5 6
8 8 9

```

Список идентификаторов:

Имя	Тип	Смысл
A	int	Исходная матрица
sum	int	Сумма элементов строки
i, j	int	Итераторы циклов
N	#define	Константа

3.5 Реализовать самостоятельно алгоритм сортировки вставками

```

#include <stdio.h>
#define SIZE 5

int main() {
    int arr[SIZE];

    printf("Enter %d elements: ", SIZE);
    for (int i = 0; i < SIZE; i++) {
        scanf("%d", &arr[i]);
    }

    for (int i = 1; i < SIZE; i++) {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }

    printf("Sorted array: ");
    for (int i = 0; i < SIZE; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}

```

Результат кода:

```
Enter 5 elements: 234
33
78
69
5
Sorted array: 5 33 69 78 234
```

Список идентификаторов:

Имя	Тип	Смысл
arr	int	Исходный массив
key	int	Вставляемый элемент
i, j	int	Итераторы циклов
SIZE	#define	Константа(размерность)