

Лабораторная работа №1

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

Простые условные конструкции. Основы работы со статическими массивами

Выполнил:

Студент 1-го курса

Гр. ИВТ-1.1

Шардт Максим

1. Комплект 1: Начала программирование. Операторы, вычисления, ВВОД-ВЫВОД.

1.1. Установить предложенную для работы среду программирования. Использовать одну из интегрированных сред разработки

Мной используется среда разработки Visual Studio code, с использованием компилятора GCC.

1.2. 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

Отсутствует.

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
input1	integer	Вводимая переменная 1
input2	integer	Вводимая переменная 2

Код программы

Lab1 - 1-2.c

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int input1, input2;
6      printf("Input 2 numbers: \n");
7      scanf("%i", &input1);
8      scanf("%i", &input2);
9      printf("Sum of your numbers is: %d\n", input1 + input2);
10
11     return 0;
12 }
```

Результат вычислений

```
Input 2 numbers:
2 2
Sum of your numbers is: 4
```

1.3. Вычислить значение выражения, 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.

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 переменной	Тип данных	Смысловое значение
----------------	------------	--------------------

x	Double	Вводимая переменная X
y	Double	Вводимая переменная Y
result	Double	Результат вычислений
denominator	Double	Знаменатель выражения
divider	Double	Делитель выражения

Код программы

```

                                Lab1 - 1-3.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  int main(void)
6  {
7      double x, y;
8      double result, denominator, divider;
9
10     printf("Input x and y: ");
11     scanf("%lf%lf", &x, &y);
12
13     denominator = 1 + sin(x + y) * sin(x + y);
14     divider = 2 + abs((x - (2 * x * x) / (1 + abs(sin(x + y)))));
15     result = denominator / divider;
16
17     printf("Result: %lf\n", result);
18
19     return 0;
20 }

```

Результат выполнения

```

Input x and y: 20 13
Result: 0.002557

```

1.4. Вычислить значение выражения. Значения параметров и аргументов можно вводить прямо в коде программы без ввода с клавиатуры.

Математическая модель

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

Выполнить для следующих значений:

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

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

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

Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
a	Float	Начальные значения
b	Float	
c	Float	
x	Float	
result	Float	Результат вычислений
denominator	Float	Знаменатель выражения
divider	Float	Делитель выражения

Код программы

Lab1 - 1-4.c

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void)
5  {
6      // float a = 0.12, b = 3.5, c = 2.4, x = 1.4;
7      // float a = 0.12, b = 3.5, c = 2.4, x = 1.6;
8      float a = 0.27, b = 3.9, c = 2.8, x = 1.8;
9
10     float result, denominator, divider;
11
12     denominator = x - a;
13     divider = pow((x * x + a * a), -3);
14     result = -denominator / divider;
15
16     denominator = 4 * pow(pow((x * x + b * b), 3), -4);
17     divider = 2 + a + b + c + pow((x - c) * (x - c), -3);
18
19     result = result - denominator / divider;
20
21     printf("%s %f\n", "Result:", result);
22
23     return 0;
24 }
25
```

Результат выполнения

Для значений $a = 0.12$, $b = 3.5$, $c = 2.4$, $x = 1.4$

```
mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
Result: -9.851794
```

Для значений $a = 0.12$, $b = 3.5$, $c = 2.4$, $x = 1.6$

```
mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
Result: -25.251657
```

Для значений $a = 0.27$, $b = 3.9$, $c = 2.8$, $x = 1.8$

```
mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
Result: -55.630932
```

2. Комплект 2: Организация циклов. Условные конструкции.

2.1. Вычислить используя цикл for координаты планеты Марс относительно Земли с течением времени t. Распечатать на экране координаты для каждой итерации по t. Подберите подходящие единицы измерения для времени и расстояния.

Математическая модель

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

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
MARS_ORBIT_RADIUS		Радиус орбиты Марса
EARTH_ORBIT_RADIUS		Радиус орбиты Земли
MARS_ORBITAL_PERIOD		Период вращения Марса
EARTH_ORBITAL_PERIOD		Период вращения Земли
MARS_W	Float	Угловая частота
EARTH_W	Float	Угловая частота Земли
x	Float	Координата X
y	Float	Координата Y

Код программы

Lab1 - 2-1.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 #define MARS_ORBIT_RADIUS 228
5 #define EARTH_ORBIT_RADIUS 150
6 #define MARS_ORBITAL_PERIOD 687
7 #define EARTH_ORBITAL_PERIOD 365
8
9 int main(void)
10 {
11     const float MARS_W = 2 * 3.14 / MARS_ORBITAL_PERIOD;
12     const float EARTH_W = 2 * 3.14 / EARTH_ORBITAL_PERIOD;
13
14     for (int time = 0; time < 30; time++)
15     {
16         float x = MARS_ORBIT_RADIUS * cos(MARS_W * time) - EARTH_ORBIT_RADIUS * cos(EARTH_W * time);
17         float y = MARS_ORBIT_RADIUS * sin(MARS_W * time) - EARTH_ORBIT_RADIUS * sin(EARTH_W * time);
18         printf("x: %f, y: %f (million km per day)\n", x, y);
19     }
20
21     return 0;
22 }
```

Результат выполнения

```
mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
x: 78.000000, y: 0.000000 (million km per day)
x: 78.012672, y: -0.496531 (million km per day)
x: 78.050697, y: -0.992473 (million km per day)
x: 78.114044, y: -1.487235 (million km per day)
x: 78.202698, y: -1.980229 (million km per day)
x: 78.316605, y: -2.470865 (million km per day)
x: 78.455719, y: -2.958558 (million km per day)
x: 78.619972, y: -3.442718 (million km per day)
x: 78.809303, y: -3.922762 (million km per day)
x: 79.023613, y: -4.398106 (million km per day)
x: 79.262802, y: -4.868166 (million km per day)
x: 79.526772, y: -5.332363 (million km per day)
x: 79.815399, y: -5.790122 (million km per day)
x: 80.128540, y: -6.240861 (million km per day)
x: 80.466072, y: -6.684009 (million km per day)
x: 80.827820, y: -7.118997 (million km per day)
x: 81.213631, y: -7.545257 (million km per day)
x: 81.623322, y: -7.962223 (million km per day)
x: 82.056709, y: -8.369334 (million km per day)
x: 82.513596, y: -8.766034 (million km per day)
x: 82.993767, y: -9.151768 (million km per day)
x: 83.497002, y: -9.525982 (million km per day)
x: 84.023071, y: -9.888138 (million km per day)
x: 84.571739, y: -10.237690 (million km per day)
x: 85.142738, y: -10.574106 (million km per day)
x: 85.735817, y: -10.896846 (million km per day)
x: 86.350700, y: -11.205386 (million km per day)
x: 86.987099, y: -11.499204 (million km per day)
x: 87.644714, y: -11.777780 (million km per day)
x: 88.323257, y: -12.040611 (million km per day)
```

2.2. Вычислить определённый интеграл от заданной функции методом трапеций. Функция $f(x)$ может быть выбрана и самостоятельно. Результат интегрирования сравнить с вычисленным вручную и убедиться в корректности результата.

Математическая модель

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

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
SUB_INTERBALS	Integer	Количество делений для вычисления интеграла
a	Double	Начальное значение интеграла
b	Double	Конечное значение интеграла
h	Double	Переменная для вычисления интеграла
sum	Double	Сумма частей интеграла
x	Double	Значение x

Код программы

Lab1 - 2-2.c

```
1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4
5  double f(double x)
6  {
7      return 1 / x;
8  }
9
10 int main(void)
11 {
12     const int SUB_INTERBALS = 1000;
13
14     double a, b;
15     printf("Enter a and b\n");
16     scanf("%lf%lf", &a, &b);
17
18     double h = fabs(b - a) / SUB_INTERBALS;
19     double sum = 0;
20
21     for (int i = 1; i < SUB_INTERBALS; i++)
22     {
23         double x = a + i * h;
24         sum = sum + f(x);
25     }
26
27     double result = (h / 2) * (f(a) + f(b) + 2 * sum);
28     printf("The integral is: %lf\n", result);
29
30     return 0;
31 }
32
```

Результат вычислений

```
mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
Enter a and b
1 10
The integral is: 2.302592
```

2.3. 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`.

Математическая модель

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

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

Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
ARRAY_SIZE	Отс.	Размер массива с числами Падована
m	Integer	Максимальное значение чисел Падована
padovan_sequenc	Integer	Массив с числами Падована

Код программы

```

Lab1 - 2-3.c

1  #include <stdio.h>
2
3  #define ARRAY_SIZE 78
4
5  int main(void)
6  {
7      int m;
8      printf("Enter m: ");
9      scanf("%d", &m);
10
11     int padovan_sequence[ARRAY_SIZE];
12     padovan_sequence[0] = padovan_sequence[1] = padovan_sequence[2] = 1;
13
14     for (int i = 3; i < ARRAY_SIZE; i++)
15     {
16         padovan_sequence[i] = padovan_sequence[i - 2] + padovan_sequence[i - 3];
17         if (padovan_sequence[i] > m)
18             break;
19     }
20
21     for (int i = 0; padovan_sequence[i] < m; i++)
22         printf("%d\n", padovan_sequence[i]);
23
24     return 0;
25 }
26

```

Результат вычислений

```

mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
Enter m: 270
1
1
1
2
2
3
4
5
7
9
12
16
21
28
37
49
65
86
114
151
200
265

```

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

Отсутствует.

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
number	Integer	Вводное число
sum	Integer	Сумма цифр числа

Код программы

Lab1 - 2-4.c

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int number;
6      int sum = 11;
7      while (sum > 10)
8      {
9          sum = 0;
10         printf("Enter a number: ");
11         scanf("%d", &number);
12         while (number > 0)
13         {
14             sum += number % 10;
15             number = number / 10;
16         }
17     }
18 }
```

Результат вычислений

```
mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
Enter a number: 999
Enter a number: 999
Enter a number: 192
Enter a number: 802
mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ █
```

3. Комплект 1: Начала программирование. Операторы, вычисления, ВВОД-ВЫВОД.

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).

Математическая модель

Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
vector_x	Integer	Вектор X
Vector_y	Integer	Вектор Y

Код программы

```

Lab1 - 3-1.c

1  #include <stdio.h>
2
3  int main(void)
4  {
5      int vector_x[2];
6      scanf("%d%d", &vector_x[0], &vector_x[1]);
7      int vector_y[2] = {vector_x[0] * vector_x[0], vector_x[1] * vector_x[1]};
8
9      printf("%s %d, %d\n", "Vector Y:", vector_y[0], vector_y[1]);
10     return 0;
11 }

```

Результат работы программы

```

mxscharde@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
10 16
Vector Y: 100, 256

```

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

Математическая модель

Отсутствует.

Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
array_size	Integer	Размер массива
initial_array	Integer	Массив чисел
reversed_array	Integer	Массив чисел с обратным порядком

Код программы

Lab1 - 3-2.c

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int array_size;
6      printf("Enter array size: ");
7      scanf("%d", &array_size);
8
9      int initial_array[array_size], reversed_array[array_size];
10     printf("Enter array: ");
11     for (int i = 0; i < array_size; i++)
12     {
13         scanf("%d", &initial_array[i]);
14         reversed_array[array_size - i - 1] = initial_array[i];
15     }
16     printf("Reversed array: ");
17     for (int i = 0; i < array_size; i++)
18         printf("%d ", reversed_array[i]);
19     printf("\n");
20
21     return 0;
22 }
```

Результат выполнения

```
mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
Enter array size: 10
Enter array: 0 1 2 3 4 5 6 7 8 9
Reversed array: 9 8 7 6 5 4 3 2 1 0
```

3.3. Транспонировать матрицу

Математическая модель

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \text{ и } \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
matrix	Integer	Изначальная матрица
matrix_t	Integer	Транспонированная матрица

Код программы

```

Lab1 - 3-3.c

1  #include <stdio.h>
2
3  int main(void)
4  {
5      int matrix[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
6      int matrix_trans[3][3];
7
8      for (int i = 0; i < 3; i++)
9          for (int j = 0; j < 3; j++)
10             matrix_trans[i][j] = matrix[j][i];
11
12     for (int i = 0; i < 3; i++)
13     {
14         for (int j = 0; j < 3; j++)
15             printf("%d ", matrix_trans[i][j]);
16         printf("\n");
17     }
18 }

```

Результат выполнения

```

mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
1 4 7
2 5 8
3 6 9

```

3.4. Преобразовать исходную матрицу так, чтобы первый элемент каждой

Математическая модель

Отсутствует.

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
MATRIX_SIZE	Отс.	Размер матрицы
matrix	Integer	Изначальная матрица
row_sum	Integer	Транспонированная матрица

Код программы

```

Lab1 - 3-4.c

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

```

Результат выполнения

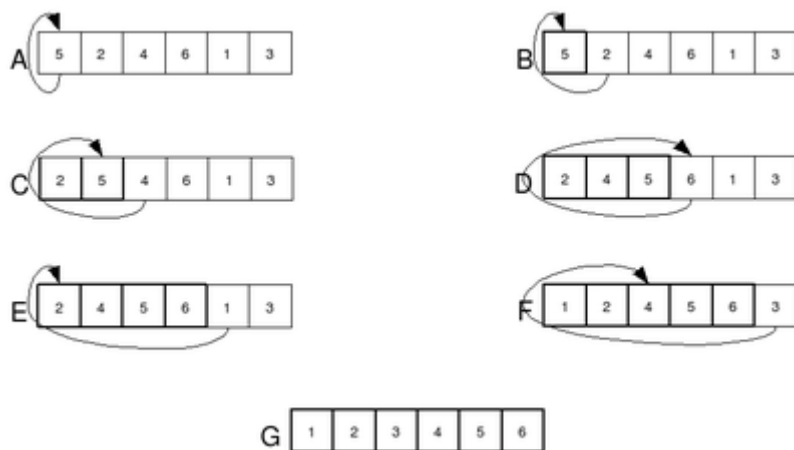
```

mxschardt@pop-os:~/Projects/C/Herzen/Lab1$ ./a.out
2 2 3
5 5 6
8 8 9

```

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

Математическая модель



Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
ARRAY_SIZE	Отс.	Размер массива
arr	Integer	Изначальный массив
temp	Integer	Временная переменная

Код программы

Lab1 - 3-5.c

```
1  #include <stdio.h>
2  #define ARRAY_SIZE 10
3
4  int main(void)
5  {
6      // Заполняем массив значениями от 100 до 1.
7      int arr[ARRAY_SIZE];
8      for (int i = ARRAY_SIZE; i > 0; i--)
9          arr[ARRAY_SIZE - i] = i;
10
11     for (int i = 1; i < ARRAY_SIZE; i++)
12         for (int j = i; (j > 0) && (arr[j - 1] > arr[j]); j--)
13             {
14                 int temp = arr[j];
15                 arr[j] = arr[j - 1];
16                 arr[j - 1] = temp;
17             }
18
19     for (int i = 0; i < ARRAY_SIZE; i++)
20         printf("%d ", arr[i]);
21     printf("\n");
22 }
```

Результат выполнения

```
/Lab1$ ./a.out
1 2 3 4 5 6 7 8 9 10
```