

Лабораторная работа №1. Задачи.

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: -

Список идентификаторов:

Имя	Тип	Смысл
A	float	1-ая вводимая переменная
B	Float	2-ая вводимая переменная
s	float	Сумма переменных a и b

Код программы:

```
#include <stdio.h>

#include <Windows.h>

#include <locale.h>

int main() {

    SetConsoleOutputCP(CP_UTF8);

    setlocale(LC_ALL, "ru_RU.UTF-8");


    float a, b;

    float s;

    printf("Ввод чисел:\n");

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

    s = a + b;

    printf("Сумма: %f", s);

    return 0;

}
```

Результат выполненной работы:

Ввод чисел:

7

8

Сумма: 15.000000

1.3

Постановка задачи:

Вычислить значение выражения:

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

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	float	1-ая вводимая переменная
Y	Float	2-ая вводимая переменная
a	float	Временная переменная
s	float	Результат вычислений

Код программы:

```
#include <stdio.h>

#include <math.h>

#include <Windows.h>

#include <locale.h>

float u(float x, float y){

    float a;

    a = sin(x + y);

    return (1 + a * a)/(2 + fabs(x - (2 * x * x) / (1 + fabs(a))));

}

int main() {

    SetConsoleOutputCP(CP_UTF8);

    setlocale(LC_ALL, "ru_RU.UTF-8");

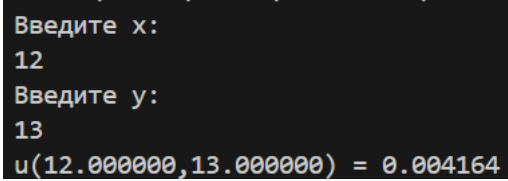
    float x, y, a, s;

    printf("Введите x:\n");

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

```
printf("Введите y:\n");  
scanf("%f", &y);  
  
s = u(x, y);  
  
printf("u(%f,%f) = %f", x,y,s);  
return 0;  
}
```

Результат выполненной работы:



```
Введите x:  
12  
Введите y:  
13  
u(12.000000,13.000000) = 0.004164
```

1.4

Постановка задачи:

Вычислить значение выражения:

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

Выполнить для следующих значений:

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

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

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

Значения параметров и аргументов можно вводить прямо в коде программы без ввода с клавиатуры.

Математическая модель:

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

Список идентификаторов:

Имя	Тип	Смысл
X	float	Переменная в выражении
A	Float	Переменная в выражении
B	float	Переменная в выражении
c	float	Переменная в выражении
M	float	Временная переменная
n	float	Временная переменная
A1,a2,a3	float	Значение по условию
B1,b2,b3	float	Значение по условию
C1,c2,c3	float	Значение по условию
H1	float	Результат вычислений
H2	float	Результат вычислений
H3	float	Результат вычислений

Код программы:

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

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

```
#include <Windows.h>
```

```
#include <locale.h>
```

```
float h(float a, float b, float c, float x) {
```

```

float m, n;

m = (x - a) / cbrt(x * x + a * a);

n = (4 * pow((x * x + b * b), 3.0 / 4.0)) / (2 + a + b + cbrt((x - c) * (x - c)));

return -m - n;

}

```

```

int main() {

    SetConsoleOutputCP(CP_UTF8);

    setlocale(LC_ALL, "ru_RU.UTF-8");


    float a1 = 0.12, b1 = 3.5, c1 = 2.4, x1 = 1.4;
    float a2 = 0.12, b2 = 3.5, c2 = 2.4, x2 = 1.6;
    float a3 = 0.27, b3 = 3.9, c3 = 2.8, x3 = 1.8;
    float h1, h2, h3;


    h1 = h(a1, b1, c1, x1);
    h2 = h(a2, b2, c2, x2);
    h3 = h(a3, b3, c3, x3);


    printf("h(%f): %f\n", x1, h1);
    printf("h(%f): %f\n", x2, h2);
    printf("h(%f): %f\n", x3, h3);


    return 0;

}

```

Результат выполненной работы:

```

h(1.400000): -5.442602
h(1.600000): -5.738755
h(1.800000): -5.992693

```

Комплект 2: Организация циклов. Условные конструкции.

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 . Координаты планеты Марс для каждой итерации задаются заданы формулами:

$$x = r_1 \cos(w_1 t) - r_2 \cos(w_2 t), \quad (3)$$

$$y = r_1 \sin(w_1 t) - r_2 \sin(w_2 t), \quad (4)$$

$$w_1 = \frac{2\pi}{T_1}, \quad (5)$$

$$w_2 = \frac{2\pi}{T_2}, \quad (6)$$

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.

Математическая модель:

$$x = r_1 * \cos\left(\frac{2\pi}{T_1} * t\right) - r_2 * \cos\left(\frac{2\pi}{T_2} * t\right)$$

$$y = r_1 * \sin\left(\frac{2\pi}{T_1} * t\right) - r_2 * \sin\left(\frac{2\pi}{T_2} * t\right)$$

Список идентификаторов:

Имя	Тип	Смысл
T1	float	Период обращения Марса
T2	Float	Период обращения Земли
R1	float	Радиус орбиты Марса
R2	float	Радиус орбиты Земли
W1	Float	Угловая скорость марса
W2	Float	Угловая скорость Земли
X	Float	Координата x
Y	Float	Координата y
t	int	Сутки

Код программы:

```
#include <stdio.h>

#include <math.h>

#include <Windows.h>

#include <locale.h>
```

```
int main() {  
    SetConsoleOutputCP(CP_UTF8);  
    setlocale(LC_ALL, "ru_RU.UTF-8");  
  
    float T1 = 687, T2 = 365, r1 = 228, r2 = 150;  
    float w1, w2, x, y;  
    float pi = 3.1415;  
  
    w1 = 2 * pi / T1;  
    w2 = 2 * pi / T2;  
  
    printf("x\ty\tcyтoк\n");  
    for (int t=0; t<=150; t+=10){  
        x = r1 * cos(w1 * t) - r2 * cos(w2 * t);  
        y = r1 * sin(w1 * t) - r2 * sin(w2 * t);  
  
        printf("%.2f\t%.2f\t%d\n", x, y, t);  
    }  
    return 0;  
}
```


Результат выполненной работы:

х	у	суток
78.00	0.00	0
79.26	-4.87	10
83.00	-9.16	20
89.03	-12.29	30
97.09	-13.75	40
106.79	-13.08	50
117.67	-9.89	60
129.19	-3.88	70
140.78	5.13	80
151.79	17.22	90
161.61	32.34	100
169.60	50.34	110
175.17	70.93	120
177.76	93.72	130
176.90	118.23	140
172.19	143.88	150

2.2

Постановка задачи:

Вычислить определённый интеграл от заданной функции методом трапеций:

$$\int_a^b f(x)dx = \int_a^b e^{x+2}dx . \quad (7)$$

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

Математическая модель:

$$\int_0^1 e^{x+2}dx$$

Список идентификаторов:

Имя	Тип	Смысл
a	double	Нижний предел интегрирования
b	double	Верхний предел интегрирования
n	Int	Точность
x	Double	Аргумент функции
h	Double	Шаг разбиения интервала
sum	Double	Накопленная сумма для метода трапеций
i	Int	Счётчик итераций цикла
result	double	Результат вычисления

Код программы:

```
#include <stdio.h>

#include <math.h>

#include <Windows.h>

#include <locale.h>

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

```
double trap(double a, double b, int n) {  
    double h = (b - a) / n;  
    double sum = (f(a) + f(b)) / 2.0;  
  
    for (int i = 1; i < n; i++) {  
        sum += f(a + i * h);  
    }  
  
    return sum * h;  
}  
  
int main() {  
    SetConsoleOutputCP(CP_UTF8);  
    setlocale(LC_ALL, "ru_RU.UTF-8");  
  
    double a = 0, b = 1;  
    int n = 1000;  
  
    double result = trap(a, b, n);  
    printf("Значение интеграла с точностью %d: %.6f\n", n, result);  
  
    return 0;  
}
```

Результат выполненной работы:

Значение интеграла С точностью 1000: 12.696482

2.3

Постановка задачи:

2.3: Организовать и распечатать последовательность чисел Падована P , 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:

$$P(0) = P(1) = P(2) = 1, \quad (8)$$

$$P(n) = P(n-2) + P(n-3). \quad (9)$$

Математическая модель:

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

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

Список идентификаторов:

Имя	Тип	Смысл
m	int	Максимальное значение числа Падована
P0	Int	Текущее значение P(n-3)
P1	Int	Текущее значение P(n-2)
p2	int	Текущее значение P(n-1)
next	int	Следующее число последовательности

Код программы:

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

```
int main() {
```

```
    int m;
```

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

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

```
    int p0 = 1, p1 = 1, p2 = 1;
```

```
    printf("1, 1, 1");
```

```
if (m == 1) {  
    return 0;  
}  
  
for (int next = p0 + p1; next <= m; ) {  
    printf(", %d", next);  
  
    p0 = p1;  
    p1 = p2;  
    p2 = next;  
    next = p0 + p1;  
}  
  
return 0;  
}
```

Результат выполненной работы:

```
Введите число m:  
300  
1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86, 114, 151, 200, 265
```

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: -

Список идентификаторов:

имя	тип	смысл
Num	int	Вводимое число
One	int	Промежуточный результат вычисления
Two	int	Промежуточный результат вычисления
Three	int	Промежуточный результат вычисления
sum	int	Результат вычислений

Код программы:

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

```
int main() {
```

```
    int num;
```

```
    while (1) {
```

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

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

```
        int one = num / 100;
```

```
        int two = (num / 10) % 10;
```

```
        int three = num % 10;
```

```
        int sum = one + two + three;
```

```
        printf("Сумма цифр: %d\n", sum);
```

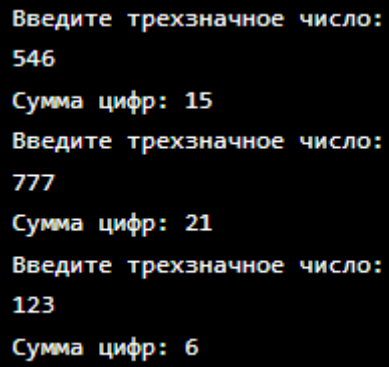
```
        if (sum <= 10) {
```

```
            break;
```

```
        }
```

```
}  
return 0;  
}
```

Результат выполненной работы:



```
Введите трехзначное число:  
546  
Сумма цифр: 15  
Введите трехзначное число:  
777  
Сумма цифр: 21  
Введите трехзначное число:  
123  
Сумма цифр: 6
```

Комплект 3: Основы работы со статическими массивами.

3.1

Постановка задачи:

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:

$$Y = X \cdot X \text{ (} y_i = x_i \cdot x_i \text{ — поэлементно).}$$

Список идентификаторов:

Имя	Тип	Смысл
size	int	Размер вектора
x	double	Вектор исходных значений
y	double	Вектор квадратов значений X
i	int	Счётчик для итерации по элементам векторов

Код программы:

```
#include <stdio.h>

int main() {
    int size;

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

    double X[size], Y[size];

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

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



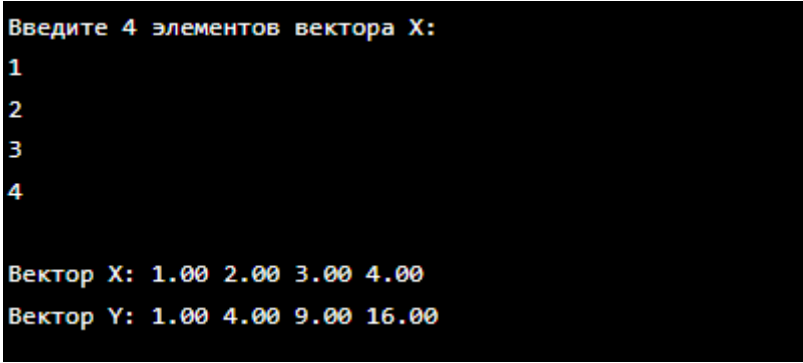
```
}

printf("\nВектор X: ");
for (int i = 0; i < size; i++) {
    printf("%.2f ", X[i]);
}

printf("\nВектор Y: ");
for (int i = 0; i < size; i++) {
    printf("%.2f ", Y[i]);
}

return 0;
}
```

Результат выполненной работы:



```
Введите 4 элементов вектора X:
1
2
3
4

Вектор X: 1.00 2.00 3.00 4.00
Вектор Y: 1.00 4.00 9.00 16.00
```

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	Смысл
size	int	Размер массива
X	double	Исходный массив элементов
i	int	Счётчик для итерации по элементам
temp	double	Временная переменная

Код программы:

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

```
int main() {
```

```
    int size;
```

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

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

```
    double X[size];
```

```
    printf("Введите %d элементов массива X:\n", size);
```

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

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

```
    }
```

```
    for (int i = 0; i < size / 2; i++) {
```

```
        double temp = X[i];
```

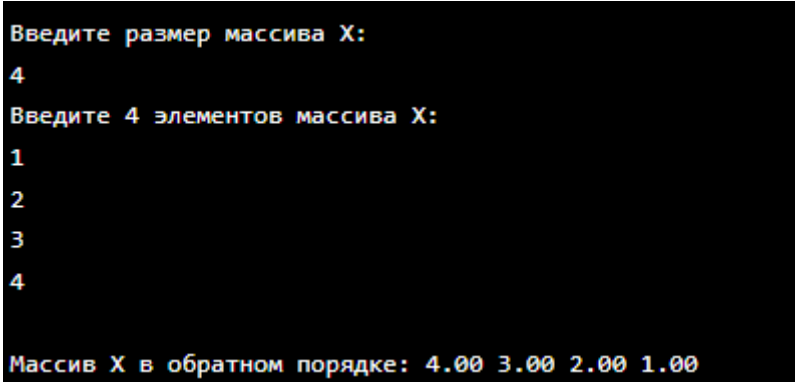
```
        X[i] = X[size - 1 - i];
```

```
        X[size - 1 - i] = temp;
```

```
    }
```

```
printf("\nМассив X в обратном порядке: ");  
for (int i = 0; i < size; i++) {  
    printf("%.2f ", X[i]);  
}  
  
return 0;  
}
```

Результат выполненной работы:



The screenshot shows a terminal window with the following text:

```
Введите размер массива X:  
4  
Введите 4 элементов массива X:  
1  
2  
3  
4  
  
Массив X в обратном порядке: 4.00 3.00 2.00 1.00
```

The program prompts the user to enter the size of array X (4), then prompts for 4 elements (1, 2, 3, 4). It then displays the array in reverse order: 4.00 3.00 2.00 1.00.

3.3

Постановка задачи:

Транспонировать матрицу:

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

Математическая модель:

$$A^T$$

Список идентификаторов:

Имя	Тип	Смысл
A	Int	Исходная матрица
B	Int	Транспонированная матрица A
i	Int	Индекс строки матрицы
j	int	Индекс столбца матрицы

Код программы:

```
#include <stdio.h>

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

    int B[3][3];

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

    printf("Исходная матрица:\n");
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
```

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

Результат выполненной работы:

Исходная матрица:

1 2 3

4 5 6

7 8 9

Транспонированная матрица:

1 4 7

2 5 8

3 6 9

3.4

Постановка задачи:

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

Математическая модель: -

Список идентификаторов:

Имя	Тип	Смысл
matrix	double	Исходная матрица
i	Int	Индекс строки матрицы
sum	Double	Сумма элементов строки
j	int	Индекс столбца матрицы

Код программы:

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

```
int main() {
```

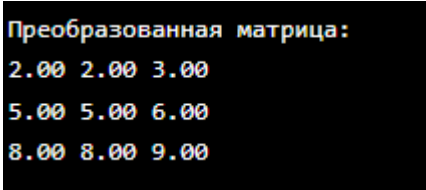
```
    double matrix[3][3] = {  
        {1, 2, 3},  
        {4, 5, 6},  
        {7, 8, 9}  
    };
```

```
    for (int i = 0; i < 3; i++) {  
        double sum = 0.0;  
        for (int j = 0; j < 3; j++) {  
            sum += matrix[i][j];  
        }  
        matrix[i][0] = sum / 3;  
    }
```

```
    printf("Преобразованная матрица:\n");  
    for (int i = 0; i < 3; i++) {  
        for (int j = 0; j < 3; j++) {
```

```
        printf("%.2f ", matrix[i][j]);  
    }  
    printf("\n");  
}  
  
return 0;  
}
```

Результат выполненной работы:



```
Преобразованная матрица:  
2.00 2.00 3.00  
5.00 5.00 6.00  
8.00 8.00 9.00
```

3.5

Постановка задачи:

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

Математическая модель: -

Список идентификаторов:

Имя	Тип	Смысл
arr	int	Исходный массив
n	Int	Количество элементов в массиве
i	Int	Индекс для сравнения элементов
k	Int	Значение элемента для вставки
j	int	Индекс текущего элемента во внешнем цикле

Код программы:

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

```
int main() {
```

```
    int arr[] = {12, 11, 13, 5, 6};
```

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

```
    printf("Исходный массив: ");
```

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

```
        printf("%d ", arr[i]);
```

```
    }
```

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

```
    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--;
```

```
        }
```

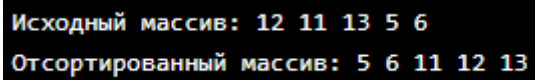
```
        arr[j + 1] = k;
```

```
    }
```



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

Результат выполненной работы:



Исходный массив: 12 11 13 5 6
Отсортированный массив: 5 6 11 12 13