

lab_1-Frolov_Andrew_1_IVT-2

Математические операции.

Переменные и их типы. Операторы. Циклы.

Простые условные конструкции.

Основы работы со статическими массивами.

Фролов Андрей, 1 курс, ИВТ-2

Комплект 1. Начало программирования. Операторы, вычисления, ввод-вывод.

Задание 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 избежание проблем с кодировками символов.

Математическая модель:

$$res = a + b$$

Список идентификаторов:

Имя	Тип	Смысл
a	int	Число 1
b	int	Число 2
res	int	Результат

Код программы:

```
1  #include <stdio.h>
2  #include <Windows.h>
3
4  int main(void){
```

```

5
6     SetConsoleOutputCP(CP_UTF8);
7
8     int a,b;
9     printf("Введите 1 число\t");
10    scanf("%d",&a);
11    printf("Введите 2 число\t");
12    scanf("%d",&b);
13    int res = a + b;
14    printf("Результат: %d",res);
15
16 }
```

Результат работы программы:

```

Введите 1 число 1
Введите 2 число 2
Результат: 3
```

Задание 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	Тип	Смысл
x	Double	Число 1
y	Double	Число 2
Chislitel	Double	Значение числителя
Znamenatel	Double	Значение знаменателя
res	Double	Результат

Код программы:

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <Windows.h>
4
5  int main(void){
6
7      SetConsoleOutputCP(CP_UTF8);
8
9      double x,y;
10     printf("Введите x\t");
11     scanf("%lf",&x);
12     printf("Введите y\t");
13     scanf("%lf",&y);
14
15     double chislitel = 1 + (sin(x+y)*sin(x+y));
16     double znamenatel = 2 + (fabs(x - (2*x*x)/(1+fabs(sin(x+y)))));
17     double res = chislitel/znamenatel;
18
19     printf("U(x,y) = %lf", res);
20 }

```

Результат работы программы:

```

Введите x      5
Введите y      10
U(x,y) = 0.052124

```

Задание 1.4.

Вычислить значение выражения для следующих значений:

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

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

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

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
a,b,c,x	Double	Заданные значения
i	int	Параметр цикла
left	Double	Левая дробь в выражении
right	Double	Правая дробь в выражении
res	Double	Результат

Код программы:

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <Windows.h>
4
5  double h(double a,double b,double c,double x){
6      double left = (x-a)/(pow(x*x+a*a,1.0/3.0));
7      double right = (4 * pow(pow(x*x+b*b,3),1.0/4.0))/(2 + a + b + pow(pow(x-
      c,2),1.0/3.0));
8      double res = -1 * left - right;
9      return res;
10 }
11
12 int main(void){
13
14     SetConsoleOutputCP(CP_UTF8);
15
16     double A[] = {0.12,0.12,0.27};
17     double B[] = {3.5,3.5,3.9};
18     double C[] = {2.4,2.4,2.8};
19     double X[] = {1.4,1.6,1.8};
20
21     for(int i = 0; i < 3; i++){
22
23         printf("%lf", h(A[i], B[i], C[i], X[i]));
24         printf("\n");
25
26     }
27 }
```

Результат работы программы:

-5.442602
-5.738755
-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 . Координаты планеты Марс для каждой итерации задаются заданы формулами:

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

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

$$w_1 = \frac{2\pi}{T_1},$$

$$w_2 = \frac{2\pi}{T_2},$$

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:

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

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

$$w_1 = \frac{2\pi}{T_1},$$

$$w_2 = \frac{2\pi}{T_2},$$

Список идентификаторов:

Имя	Тип	Смысл
r_1	double	Радиус
r_2	double	Радиус
T_1	double	Период
T_2	double	Период
w_1	double	угловая скорость?

Имя	Тип	Смысл
w_2	double	угловая скорость?
t	int	параметр цикла
x	double	координата X
y	double	координата Y

Код программы:

```

1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4  #define M_PI 3.14159265358979323846
5
6
7  int main(void)
8  {
9      SetConsoleOutputCP(CP_UTF8);
10
11     double r1 = 227.9;
12     double r2 = 149.6;
13     double T1 = 687;
14     double T2 = 365.25;
15
16     double w1 = (2*M_PI)/T1;
17     double w2 = (2*M_PI)/T2;
18
19
20     for(int t = 0; t <= 780; t+= 20)
21     {
22         double x = r1*cos(w1*t) - r2*cos(w2*t);
23         double y = r1*sin(w1*t) - r2*sin(w2*t);
24         printf("t=%d\tx=%lf\ty=%lf\n", t, x, y);
25     }
26
27 }
```

Результат работы:

t=340	x=-363.491252	y=70.250245
t=360	x=-374.300291	y=-20.768715
t=380	x=-360.129185	y=-112.225939
t=400	x=-321.801238	y=-196.790240
t=420	x=-262.368759	y=-267.729434
t=440	x=-186.803356	y=-319.593321
t=460	x=-101.479774	y=-348.756910
t=480	x=-13.511543	y=-353.763424
t=500	x=69.986444	y=-335.429596
t=520	x=142.625229	y=-296.703120
t=540	x=199.377246	y=-242.290502
t=560	x=237.049276	y=-178.099845

Задача 2.2.

Вычислить определённый интеграл от заданной функции методом трапеций:

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

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:

Имя	Тип	Смысл
$f(x)$	double	Функция для функции
$trapezoid(a, b, n)$	double	Функция для метода
r	double	результат
h	double	шаг
i	double	параметр цикла
a	double	нижний предел интегрирования
b	double	верхний предел интегрирования
n	double	количество разбиений
r	double	результат для возврата значения функции

Код программы:

```
1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4
5  double f(double x)
6  {
7      return exp(x+2);
8  }
9  double trapezoid(double a, double b, double n)
10 {
11     double r, h;
12     h = (b - a) / n;
13     r = 0;
14     for (double i = a + h; i <= b - h; i = i + h)
15     {
16         r = r + f(i);
17     }
18     return h * ((f(a) + f(b)) / 2 + r);
19 }
20
21
22 int main(void)
23 {
24     SetConsoleOutputCP(CP_UTF8);
25     double a, b, n;
26     printf("Enter the lower integration limit (a):\n");
27     scanf_s("%lf", &a);
28     printf("Enter the upper integration limit (b):\n");
29     scanf_s("%lf", &b);
30     printf("Enter the number of splits (n):\n");
31     scanf_s("%lf", &n);
32     printf("%lf", trapezoid(a,b,n));
33 }
```

Результат работы:

```
Enter the lower integration limit (a):
0
Enter the upper integration limit (b):
3
Enter the number of splits (n):
1000
140.580303
```


Задача 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`. Для этих чисел заданы формулы:

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

Математическая модель:

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

Список идентификаторов:

Имя	Тип	Смысл
P(n)	int	Ф-ция
n	int	параметр функции
m	int	число до которого выводим
i	int	параметр цикла

Код программы:

```
1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4
5  int P(int n)
6  {
7      if(n == 1 || n == 0 || n == 2)
8      {
9          return 1;
10     }
11     return P(n-2) + P(n-3);
12 }
13
14 int main(void)
15 {
16     SetConsoleOutputCP(CP_UTF8);
17 }
```

```

18     int m;
19     printf("Введите число m: ");
20     scanf("%d", &m);
21
22     printf("Последовательность чисел Падована: ");
23     for(int i = 0; P(i) <= m; i++){
24         printf("%d ", P(i));
25     }
26 }

```

Результат работы:

```

Введите число m: 200
Последовательность чисел Падована: 1 1 1 2 2 3 4 5 7 9 12 16 21 28 37 49 65 86 1
14 151 200

```

Задача 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:

$$\begin{aligned}
 sum(123) &= 1 + 2 + 3 \\
 sum > 10 &: new\ number \\
 sum \leq 10 &: stop
 \end{aligned}$$

Список идентификаторов:

Имя	Тип	Смысл
n	int	Число
sum_of_digits(n)	int	функция: сумма цифр числа
res	int	переменная для возврата результата работы функции

Код программы:

```

1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4
5  int sum_of_digits(int n)
6  {
7      int res = 0;

```

```

8     while(n>0){
9         res += n%10;
10        n/=10;
11    }
12    return res;
13 }
14
15 int main(void)
16 {
17     SetConsoleOutputCP(CP_UTF8);
18
19     int n = 999;
20     while(sum_of_digits(n)>10)
21     {
22         printf("Введите трехзначное число: ");
23         scanf("%d", &n);
24         printf("Сумма цифр числа: %d\n", sum_of_digits(n));
25         if(n>999 || n < 100)
26         {
27             printf("Введено не трехзначное число!");
28             return 0;
29         }
30     }
31 }

```

Результат работы:

```

{ gcc -std=c99 2-4.c -o 2-4 } ; if ($?) { .\2-4 }
Введите трехзначное число: 777
Сумма цифр числа: 21
Введите трехзначное число: 888
Сумма цифр числа: 24
Введите трехзначное число: 666
Сумма цифр числа: 18
Введите трехзначное число: 123
Сумма цифр числа: 6

```

Комплект 3. Основы работы со статическими массивами.

Задача 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:

$$Y = X \cdot X$$

Список идентификаторов:

Имя	Тип	Смысл
length	int	длина вектора
vector[length]	int	вектор(массив)
i	int	Параметр цикла

Код программы:

```
1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4
5  int main(void)
6  {
7
8      SetConsoleOutputCP(CP_UTF8);
9      int length;
10
11     printf("Введите длину вектора: ");
12     scanf("%d", &length);
13     int vector[length] = {};
14
15     for(int i = 0; i < length; i++)
16     {
17         printf("Введите %d число вектора: ", i);
18         scanf("%d", &vector[i]);
19         vector[i] = vector[i]*vector[i];
20     }
21
22     printf("Полученный вектор ( ");
23
24     for (int i = 0; i < length; i++)
25     {
26         printf("%d ", vector[i]);
27     }
28
29     printf(")");
30 }
```

Результат работы:

```
Введите длину вектора: 5
Введите 0 число вектора: 1
Введите 1 число вектора: 2
Введите 2 число вектора: 3
Введите 3 число вектора: 4
Введите 4 число вектора: 5
Полученный вектор ( 1 4 9 16 25 )
```

Задача 3.2.

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

Математическая модель:

$$\begin{aligned}arr[] &= (1, 2, 3) \\ newarr[] &= (3, 2, 1)\end{aligned}$$

Список идентификаторов:

Имя	Тип	Смысл
length	int	длина массива
arr[]	int	массив
i	int	параметр цикла
t	int	номер элемента с 0

Код программы:

```
1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4
5  int main(void)
6  {
7      SetConsoleOutputCP(CP_UTF8);
8
9      int length;
10
11     printf("Введите длину массива: ");
12     scanf("%d", &length);
13
14     int arr[length] = {};
15     int t = 0;
16
```

```

17     for(int i = length-1; i > -1; i--)
18     {
19         printf("Введите %d элемент массива: ", t);
20         scanf("%d", &arr[i]);
21         t++;
22     }
23
24     printf("Получившийся массив: [ ");
25     for(int i = 0; i < length; i ++ )
26     {
27         printf("%d ", arr[i]);
28     }
29     printf("]");
30
31
32 }

```

Результат работы:

```

Введите длину массива: 5
Введите 0 элемент массива: 1
Введите 1 элемент массива: 2
Введите 2 элемент массива: 3
Введите 3 элемент массива: 4
Введите 4 элемент массива: 5
Получившийся массив: [ 5 4 3 2 1 ]

```

Задача 3.3.

Транспонировать матрицу:

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

Математическая модель:

$$a[i][j] = a[j][i]$$

Список идентификаторов:

Имя	Тип	Смысл
A	int	исходная матрица
A_T	int	транспонированная матрица
i,j	int	параметры цикла

Код программы:

```
1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4
5  int main(void)
6  {
7      SetConsoleOutputCP(CP_UTF8);
8
9      int A[3][3] = { {1,2,3},{4,5,6},{7,8,9} };
10     int A_T[3][3] = { };
11
12     for(int i = 0; i < 3; i++)
13     {
14         for(int j = 0; j < 3; j++)
15         {
16             A_T[i][j] = A[j][i];
17         }
18     }
19     for(int i = 0; i < 3; i++)
20     {
21         for(int j = 0; j < 3; j++)
22         {
23             printf("%d ", A_T[i][j]);
24         }
25         printf("\n");
26     }
27 }
```

Результат работы:

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

Задача 3.4.

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

Математическая модель:

$$a[i][1] = a[1][1] + a[2][i] + \dots + a[i][j]$$

Список идентификаторов:

Имя	Тип	Смысл
A[]	int	Матрица
s	int	сумма элементов строки
rows	int	кол-во строк
cols	int	кол-во столбцов
i,j	int	параметры цикла

Код программы:

```

1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4
5  int main(void)
6  {
7      SetConsoleOutputCP(CP_UTF8);
8
9      int A[3][3] = { {11, 27, 93}, {184, 335, 16}, {71, 80, 912} };
10     int s = 0;
11     int rows = sizeof(A) / sizeof(A[0]);
12     int cols = sizeof(A[0]) / sizeof(A[0][0]);
13
14
15     printf("Исходная матрица: \n");
16     for(int i = 0; i < rows; i++)
17     {
18         for(int j = 0; j < cols; j++)
19         {
20             printf("%d ", A[i][j]);
21         }
22         printf("\n");
23     }
24
25     for (int i = 0; i < rows; i++)
26     {
27         for (int j = 0; j < cols; j++)
28         {
29             s += A[i][j];
30         }
31         A[i][0] = s/cols;
32         s = 0;
33     }

```



```

34     printf("\n");
35     printf("Полученная матрица: \n");
36     for(int i = 0; i < 3; i++)
37     {
38         for(int j = 0; j < 3; j++)
39         {
40             printf("%d ", A[i][j]);
41         }
42         printf("\n");
43     }
44
45
46 }

```

Результат работы:

```

Исходная матрица:
11 27 93
184 335 16
71 80 912

Полученная матрица:
43 27 93
178 335 16
354 80 912

```

Задача 3.5.

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

Математическая модель:

Входные данные: $A = [a_1, a_2, \dots, a_n]$.

Для каждого $i \in \{2, 3, \dots, n\}$:

Найти $j \in \{1, 2, \dots, i\}$ такое, что $a_{j-1} \leq a_i < a_j$.

Сдвинуть элементы $[a_j, a_{j+1}, \dots, a_{i-1}]$ вправо на одну позицию.

Поместить a_i на позицию j .

Итоговый результат: $A = [a_1, a_2, \dots, a_n]$, где $a_1 \leq a_2 \leq \dots \leq a_n$.

Список идентификаторов:

Имя	Тип	Смысл
arr[]	int	массив

Имя	Тип	Смысл
temp	int	переменная для перестановки элементов
length	int	длина массива
i,j	int	параметры цикла

Код программы:

```

1  #include <math.h>
2  #include <stdio.h>
3  #include <Windows.h>
4
5  int main(void)
6  {
7      SetConsoleOutputCP(CP_UTF8);
8
9      int arr[] = {42, 17, 89, 5, 23, 56, 34, 72, 11, 63, 28, 95, 50, 7, 39};
10     int temp;
11     int length = sizeof(arr)/sizeof(int);
12
13     printf("\nИсходная матрица: \n");
14
15     for (int i = 0; i < length; i++)
16     {
17         printf("%d ", arr[i]);
18     }
19
20
21     for (int i = 1; i < length; i++)
22     {
23         for (int j = i; j > 0 && arr[j-1]>arr[j]; j--)
24         {
25             temp = arr[j-1];
26             arr[j-1] = arr[j];
27             arr[j] = temp;
28         }
29     }
30
31     printf("\nПолученная матрица: \n");
32     for(int i = 0; i < length; i++)
33     {
34         printf("%d ", arr[i]);
35     }
36 }

```

Результат работы:

Исходная матрица:

42 17 89 5 23 56 34 72 11 63 28 95 50 7 39

Полученная матрица:

5 7 11 17 23 28 34 39 42 50 56 63 72 89 95