

## Лабораторная работа №2

### Задание 2.1

#### Постановка задачи

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:

$$x = r_1 \cos(w_1 t) - r_2 \cos(w_2 t), \quad (1)$$

$$y = r_1 \sin(w_1 t) - r_2 \sin(w_2 t), \quad (2)$$

$$w_1 = \frac{2\pi}{T_1}, \quad (3)$$

$$w_2 = \frac{2\pi}{T_2}, \quad (4)$$

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

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

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

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

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

### Описание переменных

Переменная	Тип	Суть
x	double	Х-координата Марса
y	double	Y-координата Марса
t	double	День
t1	double	Период обращения Марса в днях
t2	double	Период обращения Земли в днях
w1	double	Переменная $w_1$ из мат. модели
w2	double	Переменная $w_2$ из мат. модели
r1	double	Расстояние от Марса до Солнца в км
r2	double	Расстояние от Земли до Солнца в км

### Код программы

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void main() {
5      double x, y, t, t1, t2, w1, w2, r1, r2;
6      t1 = 687; // период обращения Марса в днях
7      t2 = 365.25; // период обращения Земли в днях
8      r1 = 227900000; // расстояние от Марса до Солнца в км
9      r2 = 149600000; // расстояние от Земли до Солнца в км
10     w1 = 2 * M_PI / t1;
11     w2 = 2 * M_PI / t2;
12
13     for (t = 0; t <= 365; ++t) {
14         x = r1 * cos(w1 * t) - r2 * cos(w2 * t);
15         y = r1 * sin(w1 * t) - r2 * sin(w2 * t);
16         printf("X = %-12lg Y = %-12lg\n", x, y);
17     }
18 }
19
```

### Вывод программы

```
X = 7.83e+07      Y = 0
X = 7.83126e+07  Y = -489050
X = 7.83504e+07  Y = -977513
X = 7.84134e+07  Y = -1.4648e+06
X = 7.85015e+07  Y = -1.95033e+06
X = 7.86148e+07  Y = -2.43351e+06
X = 7.87531e+07  Y = -2.91376e+06
X = 7.89164e+07  Y = -3.3905e+06
X = 7.91047e+07  Y = -3.86313e+06
X = 7.93177e+07  Y = -4.33109e+06
X = 7.95556e+07  Y = -4.79379e+06
X = 7.9818e+07   Y = -5.25065e+06
X = 8.0105e+07   Y = -5.7011e+06
X = 8.04163e+07  Y = -6.14456e+06
X = 8.07519e+07  Y = -6.58046e+06
X = 8.11116e+07  Y = -7.00825e+06
X = 8.14952e+07  Y = -7.42734e+06
X = 8.19025e+07  Y = -7.83718e+06
...
```

## Задание 2.2

### Постановка задачи

2.2: Вычислить определённый интеграл от заданной функции методом трапеций:

$$\int_a^b f(x)dx = \int_a^b e^{x+2}dx. \quad (5)$$

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

### Математическая модель

$$\int_a^b \frac{\cos(x)dx}{x}$$

### Описание переменных

Переменная	Тип	Суть
a	double	Нижний предел интегрирования
b	double	Верхний предел интегрирования
x	double	Аргумент функции
f	double	Среднее значение подынтегральной функции
f1	double	Значение подынтегральной функции в точке X
f2	double	Значение подынтегральной функции в точке $X + h$
integral	double	Искомый интеграл
h	double	Шаг
n	int	Количество разбиений

### Код программы

```
1  #include <stdio.h>
2  #include <math.h>
3
4  // подынтегральная функция
5  double func(double x) {
6      return cos(x) / x;
7  }
8
9  void main() {
10     double a, b, f, x, f1, f2, h, integral = 0;
11     int n = 10000;
12     printf("a, b: ");
13     scanf("%lg %lg", &a, &b);
14     h = (b - a) / n;
15     x = a;
16
17     while (x < b) {
18         f1 = func(x);
19         x += h;
20         f2 = func(x);
21         f = (f1 + f2) / 2;
22         integral += f;
23     }
24
25     integral *= h;
26
27     printf("Интеграл = %lg", integral);
28 }
29
```

## Вывод программы

a, b: 0.5 4  
Интеграл = 0.0368024

## Проверка правильности



[Extended Keyboard](#) [Upload](#) [Examples](#) [Random](#)

Definite integral:

$$\int_{0.5}^4 \frac{\cos(x)}{x} dx = 0.0368024$$

Visual representation of the integral:

Indefinite integral:

$$\int \frac{\cos(x)}{x} dx = \text{Ci}(x) + \text{constant}$$

Ci(x) is the cosine integral

[Download Page](#) POWERED BY THE WOLFRAM LANGUAGE

## Задание 2.3

### Постановка задачи

2.3: Организовать и распечатать последовательность чисел Падована<sup>[1]</sup>, 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:

$$P(0) = P(1) = P(2) = 1, \quad (6)$$

$$P(n) = P(n - 2) + P(n - 3). \quad (7)$$

### Математическая модель

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

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

### Описание переменных

Переменная	Тип	Суть
a	Array of int	Массив элементов последовательности от $n - 3$ до $n$
m	int	Ограничение значения эл-та
p	int	Текущий эл-т последовательности

### Код программы

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

### Вывод программы

m = 321

1 1 1 2 2 3 4 5 7 9 12 16 21 28 37 49 65 86 114 151 200 265



## Задание 2.4

### Постановка задачи

2.4: Дано натуральное число. Вывести на экран все натуральные числа до заданного включительно.

### Описание переменных

Переменная	Тип	Суть
n	int	Счетчик цикла
i	int	Верхняя граница диапазона

### Код программы

```
1  #include <stdio.h>
2
3  void main() {
4      printf("Верхняя граница диапазона: ");
5      int i;
6      scanf("%d", &i);
7
8      for (int n = 0; n <= i; ++n) {
9          printf("%d ", n);
10     }
11 }
12
```

### Вывод программы

Верхняя граница диапазона: 15

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 Задание 2.5

## Задание 2.5

### Постановка задачи

2.5: 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.

### Описание переменных

Переменная	Тип	Суть
dig1	int	Цифра 1
dig2	int	Цифра 2
dig3	int	Цифра 3

### Код программы

```
1  #include <stdio.h>
2
3  void main() {
4      int dig1, dig2, dig3;
5
6      while (1) {
7          printf("Введите 3-х значное число: ");
8          scanf("%1d%1d%1d", &dig1, &dig2, &dig3);
9          if ((dig1 + dig2 + dig3) <= 10) break;
10     }
11
12     printf("Работа завершена");
13 }
14
```

### **Вывод программы**

Введите 3-х значное число: 555  
Введите 3-х значное число: 666  
Введите 3-х значное число: 654  
Введите 3-х значное число: 111  
Работа завершена