

## Комплект 1: Начала программирования. Операторы, вычисления, ввод-вывод.

### Задача 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

$$\text{sum} = a + b$$

#### Список идентификаторов

Имя	Тип	Значение
a, b	int	Вводимые числа

#### Код программы на языке C

```
1  #include <stdio.h>
2
3
4  int main(void) {
5      int a, b;
6
7      printf("\nPrivet! Invite you to input 2 numbers!\n");
8      scanf("%d%d", &a, &b);
9      printf("\nOgo! Sum is %d\n", a + b);
10 }
```

#### Результат работы программы

```
Privet! Invite you to input 2 numbers!
10
20

Ogo! Sum is 30
```

### Задача 1.3

**Постановка задачи:** Вычислить значение выражения  $u(x, y)$ , 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	Значение
u	float	Функция $u(x, y)$
x, y	float	Вводимые числа $x$ и $y$
sinxy	float	$\sin(x + y)$

#### Код программы на языке C

```
1  #include <stdio.h>
2  #include <math.h>
3
4
5  float u(float x, float y) {
6      float sinxy = sinf(x + y);
7      return (1 + powf(sinxy, 2)) / (2 + fabs(x - (2*x*x / (1 + fabs(sinxy)))));
8  }
9
10 int main(void) {
11     float x, y;
12
13     printf("\nPrivet! Invite you to input 2 numbers: x and y!\n");
14     scanf("%f%f", &x, &y);
15     printf("\nWow! u(x, y) is %f\n\n", u(x, y));
16 }
```

#### Результат работы программы

```
Privet! Invite you to input 2 numbers: x and y!
-1
1

Wow! u(x, y) is 0.200000
```

## Задача 1.4

**Постановка задачи:** Вычислить значение выражения  $h(x)$  для приведенных ниже значений.

**Математическая модель**

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

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

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

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

**Список идентификаторов**

Имя	Тип	Значение
h	double	Функция h(a, b, c, x)
a, b, c, x	double	Аргументы функции
term1, term2	double	Слагаемые в выражении
numerator	double	Числитель дроби во втором слагаемом
denominator	double	Знаменатель дроби во втором слагаемом

**Код программы на языке C**

```

1  #include <stdio.h>
2  #include <math.h>
3
4
5  double h(double a, double b, double c, double x) {
6      double term1 = -(x - a) / pow(x * x + a * a, 1.0 / 3.0);
7      double numerator = 4.0 * pow(x * x + b * b, 3.0 / 4.0);
8      double denominator = 2.0 + a + b + pow((x - c) * (x - c), 1.0 / 3.0);
9      double term2 = -numerator / denominator;
10     return term1 + term2;
11 }
12
13 int main(void) {
14     double a = 0.12, b = 3.5, c = 2.4, x = 1.4;
15     printf("\nh(%f, %f, %f, %f) = %f\n", a, b, c, x, h(a, b, c, x));
16
17     a = 0.12; b = 3.5; c = 2.4; x = 1.6;
18     printf("h(%f, %f, %f, %f) = %f\n", a, b, c, x, h(a, b, c, x));
19
20     a = 0.27; b = 3.9; c = 2.8; x = 1.8;
21     printf("h(%f, %f, %f, %f) = %f\n", a, b, c, x, h(a, b, c, x));
22
23     return 0;
24 }
```

**Результат работы программы**

```

h(0.120000, 3.500000, 2.400000, 1.400000) = -5.442602
h(0.120000, 3.500000, 2.400000, 1.600000) = -5.738755
h(0.270000, 3.900000, 2.800000, 1.800000) = -5.992693
```

## Комплект 2: Организация циклов. Условные конструкции.

### Задача 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:

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

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

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

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

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 повremени. Подберите подходящие единицы измерения для времени и расстояния.

### Список идентификаторов

Имя	Тип	Значение
$r_1, r_2, t_1, t_2$	const double	Исходные значения в СИ
$w_1, w_2$	double	Угловые скорости планет
$t$	int	Счётчик цикла, время в днях
$t\_sec$	double	Время в СИ
$x, y$	double	Координаты Марса относительно земли

### Код программы на языке C

```
1  #include <stdio.h>
2  #include <math.h>
3
4
5  int main(void) {
6      const double r1 = 228e9;
7      const double r2 = 150e9;
8      const double t1 = 686.96 * 24 * 3600;
9      const double t2 = 365.26 * 24 * 3600;
10
11     double w1 = (2 * M_PI) / t1;
12     double w2 = (2 * M_PI) / t2;
13
14     printf("\nt, days\ttx, mln. km\tty, mln. km\n");
15     for (int t = 0; t < 365; t++) {
16         double t_sec = t * 24 * 3600;
17         double x = r1 * cos(w1 * t_sec) - r2 * cos(w2 * t_sec);
18         double y = r1 * sin(w1 * t_sec) - r2 * sin(w2 * t_sec);
19
20         printf("%d\t%.4f\t%.4f\n", t + 1, x / 1e9, y / 1e9);
21     }
22
23     return 0;
24 }
```

**Результат работы программы**

t, days	x, mln. km	y, mln. km
1	78.0000	0.0000
2	78.0127	-0.4948
3	78.0506	-0.9891
4	78.1139	-1.4821
5	78.2024	-1.9734
6	78.3161	-2.4623
7	78.4550	-2.9483
8	78.6190	-3.4308
9	78.8080	-3.9092
10	79.0220	-4.3828
11	79.2608	-4.8512
12	79.5244	-5.3138
13	79.8125	-5.7699
14	80.1252	-6.2189
15	80.4622	-6.6605
16	80.8234	-7.0938
17	81.2086	-7.5185
18	81.6176	-7.9338
19	82.0503	-8.3393
20	82.5065	-8.7345
21	82.9859	-9.1186
22	83.4884	-9.4913
23	84.0136	-9.8519
24	84.5614	-10.2000
25	85.1315	-10.5349
26	85.7236	-10.8561
27	86.3375	-11.1632
28	86.9729	-11.4556
29	87.6295	-11.7327
30	88.3070	-11.9942
31	89.0050	-12.2393
32	89.7233	-12.4678
33	90.4614	-12.6790
34	91.2192	-12.8725
35	91.9962	-13.0479
36	92.7920	-13.2045
37	93.6063	-13.3420
38	94.4387	-13.4600
39	95.2887	-13.5578
40	96.1561	-13.6352
41	97.0404	-13.6917
42	97.9412	-13.7267
43	98.8580	-13.7400

---

**Задача 2.2**

**Постановка задачи:** Вычислить определённый интеграл от заданной функции методом трапеций:

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

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

**Математическая модель**

$$I = \int_a^b f(x) dx \approx h \left( \frac{1}{2} f(x_0) + f(x_1) + \dots + f(x_{n-1}) + \frac{1}{2} f(x_n) \right) = h \left( \frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right),$$

где  $x \in [a; b]$ ,  $\Delta x = h$

**Список идентификаторов**

Имя	Тип	Значение
n	int	Число итераций
a, b	int	Границы интегрирования
h	double	Шаг интегрирования
x	double	Переменная интегрирования
I	double	Вычисляемый интеграл
f	double	Вычисление интегрируемой функции

**Код программы на языке C**

```
1  #include <stdio.h>
2  #include <math.h>
3
4
5  int main(void) {
6      int n = 100000;
7      double a = 0;
8      double b = 5;
9
10     double h = (b - a) / n;
11     double I = 0;
12
13     for (double x = a; x <= b; x += h) {
14         double f = exp(x + 2.0);
15         f = (x == a || x == b) ? f/2 : f;
16         I += f;
17     }
18     I *= h;
19     printf("\nI = %f\n\n", I);
20 }
```

**Результат работы программы**

I = 1089.271518

### Задача 2.3

**Постановка задачи:** 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`. Для этих чисел заданы формулы:

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

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

#### Список идентификаторов

Имя	Тип	Значение
P	int	Рекурсивная функция
n	int	Аргумент функции
m	int	Число m
p	int	Результат вычисления функции P(n) в цикле по n

#### Код программы на языке C

```
1  #include <stdio.h>
2
3
4  int P(int n) {
5      if (n > 2) {
6          return P(n - 2) + P(n - 3);
7      } else {
8          return 1;
9      }
10 }
11
12 int main(void) {
13     int m;
14     int p = 1;
15
16     printf("\nPrivet! Invite you to input m: ");
17     scanf("%d", &m);
18
19     printf("Padovan numbers:\n");
20     for (int n = 0; p <= m; n++) {
21         printf("%d\n", p);
22         p = P(n);
23     }
24 }
```

**Результат работы программы**

```
Privet! Invite you to input m: 100
Padovan numbers:
1
1
1
1
2
2
3
4
5
7
9
12
16
21
28
37
49
65
86
```

---

**Задача 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.

**Список идентификаторов**

Имя	Тип	Значение
a	int	Вводимое число
sum	int	Сумма цифр
i	int	Счётчик цикла



**Код программы на языке C**

```
1  #include <stdio.h>
2
3
4  int main(void) {
5      int a;
6      int sum = 11;
7
8      printf("\nPrivet! Invite you to input 3-diget number:\n");
9      for (int i = 0; sum > 10; i++) {
10         scanf("%d", &a);
11         sum = (a / 100) + (a % 100 / 10) + (a % 10);
12     }
13     printf("\n");
14
15     return 0;
16 }
```

**Результат работы программы**

```
Privet! Invite you to input 3-diget number:
444
551
900
```

**Комплект 3: Основы работы со статическими массивами.****Задача 3.1**

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

**Список идентификаторов**

Имя	Тип	Значение
n	int	Длина вектора
x[n], y[n]	int	Массивы для векторов X и Y
i	int	Счётчик цикла

**Код программы на языке C**

```
1  #include <stdio.h>
2
3
4  int main(void) {
5      int n;
6      printf("\nInput vector X length: ");
7      scanf("%d", &n);
8
9      int x[n], y[n];
10     for (int i = 0; i < n; i++) {
11         printf("Input x_%d: ", i);
12         scanf("%d", &x[i]);
13         y[i] = x[i] * x[i];
14     }
15
16     printf("\nVector Y:\n");
17     for (int i = 0; i < n; i++) {
18         printf("y_%d: %d\n", i, y[i]);
19     }
20     printf("\n");
21
22     return 0;
23 }
```

**Результат работы программы**

```
Input vector X length: 5
Input x_0: 1
Input x_1: 2
Input x_2: 3
Input x_3: 4
Input x_4: 5

Vector Y:
y_0: 1
y_1: 4
y_2: 9
y_3: 16
y_4: 25
```

### Задача 3.2

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

#### Список идентификаторов

Имя	Тип	Значение
n	int	Длина массива
x[n]	int	Массив X
i	int	Счётчик цикла

#### Код программы на языке C

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

#### Результат работы программы

```
Input array X length: 6
Input x_0: 1
Input x_1: 2
Input x_2: 3
Input x_3: 4
Input x_4: 5
Input x_5: 6

Reverse array X:
x_0: 6
x_1: 5
x_2: 4
x_3: 3
x_4: 2
x_5: 1
```

### Задача 3.3

**Постановка задачи:** Транспонировать матрицу:

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

#### Список идентификаторов

Имя	Тип	Значение
a	int	Исходная матрица
b	int	Транспонированная матрица
i	int	Счётчик цикла

#### Код программы на языке C

```
1  #include <stdio.h>
2
3
4  int main(void) {
5      int a[3][3] = {
6          {1, 2, 3},
7          {4, 5, 6},
8          {7, 8, 9}
9      };
10     int b[3][3];
11
12     printf("\nMatrix A transposed:\n");
13     for (int i = 0; i < 3; i++) {
14         for (int j = 0; j < 3; j++) {
15             b[i][j] = a[j][i];
16             printf("%d ", b[i][j]);
17         }
18         printf("\n");
19     }
20     printf("\n");
21
22     return 0;
23 }
```

#### Результат работы программы

```
Matrix A transposed:
1 4 7
2 5 8
3 6 9
```

### Задача 3.4

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

#### Список идентификаторов

Имя	Тип	Значение
a	double	Исходная матрица
b	double	Преобразованная матрицы
I, J	const int	Размеры матрицы
i, j	int	Счётчики циклов
sum	double	Сумма элементов строки

#### Код программы на языке C

```
1  #include <stdio.h>
2
3
4  int main(void) {
5      const int I = 4;
6      const int J = 3;
7      double a[I][J] = {
8          {1, 2, 3},
9          {4, 5, 6},
10         {7, 8, 9},
11         {10, 11, 12}
12     };
13
14     for (int i = 0; i < I; i++) {
15         double sum = 0;
16         for (int j = 0; j < 3; j++) {
17             sum += a[i][j];
18         }
19         a[i][0] = sum / J;
20     }
21
22     printf("\nMatrix B:\n");
23     for (int i = 0; i < I; i++) {
24         for (int j = 0; j < 3; j++) {
25             printf("%.3f\t", a[i][j]);
26         }
27         printf("\n");
28     }
29     printf("\n");
30
31     return 0;
32 }
```

#### Результат работы программы

```
Matrix B:
2.000  2.000  3.000
5.000  5.000  6.000
8.000  8.000  9.000
11.000 11.000 12.000
```

### Задача 3.5

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

#### Список идентификаторов

Имя	Тип	Значение
n	const int	Длина массива
a[n]	int	Исходный массив
buf	int	Буфер обмена
i, j	int	Счётчики циклов

#### Код программы на языке C

```
1  #include <stdio.h>
2
3
4  int main(void) {
5      const int n = 7;
6      int a[n] = {1, 30, 14, 5, 90, 1, 10};
7      int buf;
8
9      for(int i = 1; i < n; i++) {
10         for(int j = i; j > 0 && a[j-1] > a[j]; j--) {
11             buf = a[j-1];
12             a[j-1] = a[j];
13             a[j] = buf;
14         }
15     }
16
17     printf("\nSorted array: ");
18     for (int i = 0; i < n; i++) {
19         printf("%d ", a[i]);
20     }
21
22     return 0;
23 }
```

#### Результат работы программы

```
Sorted array: 1 1 5 10 14 30 90
```