

Технология программирования на ЭВМ

Функции

Баев А.Ж.

Казахстанский филиал МГУ

29 ноября 2018

Пример: дискриминант

Описание функции:

```
1 double discr(double a, double b, double c) {  
2     double d = b * b - 4.0 * a * c;  
3     return d;  
4 }
```

Вызов функции:

```
1 int main() {  
2     double d1, d2, d3;  
3     double a = 2.0, b = 3.0, c = 1.0;  
4     d1 = discr(2.0, 5.0, 3.0);  
5     d2 = discr(a, b, c);  
6     d3 = discr(2.0, 5.0, c);  
7     return 0;  
8 }
```

Что дают функции?

- ❶ нет длинных блоков кода;
- ❷ нет повторяющегося кода;
- ❸ проще читать код.

Можно сначала описать только заголовок функции:
возвращаемый тип, имя и аргументы.

```
1 double discr(double a, double b, double c);
```

А ниже (обычно после main) описать реализацию:

```
1 double discr(double a, double b, double c) {  
2     return b * b - 4 * a * c;  
3 }
```

Без аргументов

Функция без аргументов

```
1 double readDouble(void) {
2     double x;
3     scanf("%lf", &x);
4     return x;
5 }
6 int main () {
7     double a;
8     a = readDouble();
9     return 0;
10 }
```

Также можно использовать просто пустые скобки:

```
1 double readDouble() {
2     ...
3 }
```

Без возвращаемого значения

Функция без аргументов

```
1 void hello(int n) {  
2     int i;  
3     for (i = 0; i < n; i++) {  
4         puts("hello");  
5     }  
6     return;  
7 }  
8 int main () {  
9     hello(5);  
10    return 0;  
11 }
```

Оператор return; можно опустить.

Без возвращаемого значения и аргументов

```
1 void make(void) {  
2     int a, b;  
3     scanf("%d %d", &a, &b);  
4     printf("%d\n", a + b);  
5 }  
6  
7 int main() {  
8     make();  
9     return 0;  
10 }
```

Длина вектора

Функция: найти длину вектора.

```
1 int main() {  
2     double a = 3.0, b = 4.0, l1, l2, l3;  
3     l1 = len(5.0, 12.0);  
4     l2 = len(a, b);  
5     l3 = len(a, 3.0);  
6     return 0;  
7 }
```


Длина вектора

```
1 double len(double x, double y) {  
2     return sqrt(x * x + y * y);  
3 }
```

Количество делителей

Функция: найти количество делителей числа.

```
1  int main() {  
2      int m = 4, a, b;  
3      a = tau(10);  
4      b = tau(m);  
5      return 0;  
6  }
```

Количество делителей

```
1 int tau(int n) {  
2     int ans = 0, d;  
3     for (d = 1; d <= n; d++) {  
4         if (n % d == 0) {  
5             ans++;  
6         }  
7     }  
8     return ans;  
9 }
```

Функция: определить знак вещественного числа (1 для положительных, -1 для отрицательных, 0 для нуля).

```
1  int main() {  
2      int a, b, c;  
3      a = sign(5.1);  
4      b = sign(-4.2);  
5      c = sign(0.0);  
6      return 0;  
7  }
```

```
1 int sign(double x) {  
2     if (x > 0) {  
3         return 1;  
4     }  
5     if (x < 0) {  
6         return -1;  
7     }  
8     return 0;  
9 }
```

Функция: определить максимум из двух целых чисел.

```
1 int main() {  
2     int a, b;  
3     a = max(2, 5);  
4     b = max(max(2, 3), 5);  
5     c = max(2, max(3, 5));  
6     return 0;  
7 }
```

Максимум — с предупреждением

```
1 int max(int a, int b) {  
2     if (a > b) {  
3         return a;  
4     } else {  
5         return b;  
6     }  
7 }
```

Warning: no `return` value by default

Максимум — без предупреждения

```
1 int max(int a, int b) {  
2     if (a > b) {  
3         return a;  
4     }  
5     return b;  
6 }
```


Функция: определить максимум из трех целых чисел.

```
1 int main() {  
2     int a;  
3     a = max3(2, 5, 3);  
4     return 0;  
5 }
```

Максимум

```
1 int max3(int a, int b, int c) {  
2     return max(max(a, b), c);  
3 }
```

Передача по значению

Что будет в переменных a и b?

```
1 void next(int d) {  
2     d = d + 1;  
3 }  
4  
5 int main() {  
6     int x = 6, a = 0, b = 0;  
7     a = x;  
8  
9     next(x);  
10    b = x;  
11  
12    return 0;  
13 }
```

Почему?

$x = 6 \mid a = 0 \mid b = 0$ `int x = 6, a = 0, b = 0;`

$x = 6 \mid a = 6 \mid b = 0$ `a = x;`

$x = 6 \mid a = 6 \mid b = 0$ `next(x);`

$x = 6 \mid a = 6 \mid b = 0$

$x = 6 \mid a = 6 \mid b = 0$

$x = 6 \mid a = 6 \mid b = 6$ `b = x;`

$d = 6$ `void next(int d)`

$d = 7$ `d = d + 1;`

Передача по указателю

Что будет в переменных a и b?

```
1 void next(int *d) {  
2     *d = *d + 1;  
3 }  
4  
5 int main() {  
6     int x = 6, a = 0, b = 0;  
7     a = x;  
8  
9     next(&x);  
10    b = x;  
11  
12    return 0;  
13 }
```

Почему?

x = 6 | a = 0 | b = 0 int x = 6, a = 0, b = 0;

x = 6 | a = 6 | b = 0 a = x;

x = 6 | a = 6 | b = 0 next(&x);

x = 6 | a = 6 | b = 0

d = 11480 void next(int *d)

x = 7 | a = 6 | b = 0

d = 11480 *d = *d + 1;

x = 7 | a = 6 | b = 7 b = x;

Массив как аргумент

Функция: посчитать сумму элементов массива.

```
1  int main() {  
2      int a[5] = {1, 5, 2, 4, 3}, answer;  
3      answer = sum(5, a);  
4      return 0;  
5  }
```

Массив как аргумент

```
1  int sum(int n, int a[]) {  
2      int s = 0, i;  
3      for (i = 0; i < n; i++) {  
4          s += a[i];  
5      }  
6      return s;  
7  }
```

Можно использовать следующий прототип:

```
1  int sum(int n, int *a) {  
2      ...  
3  }
```


Функция Эйлера

Дано целое n от 2 до 10^6 . Вычислить $\varphi(n)$ — количество натуральных чисел меньших n и взаимнопростых с n .

Ввод

10

Вывод

4

`int gcd(int a, int b)` — вычисляет наибольший общий делитель чисел a и b по алгоритму Евклида;
`void phi(int n)` — вычисляет $\varphi(n)$ (подсчитывает количество чисел a таких, что $(a, n) = 1$).

Функция Эйлера

```
1  #include <stdio.h>
2  int gcd(int a, int b) {
3      while (b != 0) {
4          t = a % b;
5          a = b;
6          b = t;
7      }
8      return a;
9  }
10 int phi(int n) {
11     int answer = 0, a;
12     for (a = 1; a < n; a++)
13         if (gcd(a, n) == 1)
14             answer++;
15     return answer;
16 }
```

Функция Эйлера

```
17 int main() {  
18     int n;  
19     scanf("%d", &n);  
20     printf("%d\n", phi(n));  
21     return 0;  
22 }
```