Технология программирования на ЭВМ
Цикл while

Баев А.Ж.

Казахстанский филиал МГУ

06 ноября 2019

Пример. Степень двойки.

Дано целое положительное число n от 1 до 15. Найти минимальную степень двойки, которая больше n.

Ввод	5	3	8
Вывод	8	4	16

Решение if

Будем перебирать все степени двойки 2, 4 и 8. Если степень всё еще меньше, то её можно еще увеличить.

```
 1    int p = 2, n;
 2    scanf("%d", &n);
 3    if (p <= n) {
 4        p *= 2;
 5    }
 6    if (p <= n) {
 7        p *= 2;
 8    }
 9    if (p <= n) {
10        p *= 2;
11    }
12    printf("%d", n);
```

Решение while

Будем перебирать все степени двойки 2, 4 и 8. Если степень всё еще меньше, то её можно еще увеличить.

```
1     int p = 2, n;
2     scanf("%d", &n);
3     while (p <= n) {
4         p *= 2;
5     }
6     printf("%d", n);
```

Скобки — не обязательны.

Общий вид

Тело цикла body выполняется если условие condition верно.

```
1   while (condition) {
2       body;
3   }
```

Пример. Количество нулей.

Дано целое положительно число n от 1 до 10^6. Необходимо посчитать, на сколько нулей оканчивается число.

Ввод	123000	1000000	2001
Вывод	3	6	0

Решение

```
1       int ans = 0, n;
2       scanf("%d", &n);
3
4       while (n % 10 == 0) {
5           n /= 10;
6           ans++;
7       }
```

Скобки — обязательны.

Что будет в неправильном решении?

```
1    int ans = 0, n;
2    scanf("%d", &n);
3
4    while (n % 10 == 0)
5        n /= 10;
6        ans++;
```

Зацикливание

Цикл не останавливается.

```
1     int ans = 0, n = 100;
2
3     while (n % 10 == 0)
4         ans++;
5         n /= 10;
```

В таких случае рекомендуется снять процесс комбинацией: `Ctrl + C`.

Эхо.

Программа выводит введенные числа на экран до тех пор, пока не встретится нуль.

Ввод	10 20 0
Вывод	10 20

Решение без вывода нуля.

```
1    int a;
2    scanf("%d", &a);
3    while(a != 0) {
4        printf("%d", a);
5        scanf("%d", &a);
6    }
```

Решение с выводом нуля.

```
1     int a;
2     do {
3         scanf("%d", &a);
4         printf("%d", a);
5     } while(a != 0);
```

Цикл с постусловием do while.

Тело цикла body выполняется после чего, проверяется условие condition. Если оно верно, то повторяем действия.

```
1     do {
2         body;
3     } while (condition);
```

Переменная–счетчик.

Вывести текст «HELLO!» 5 раз.

```
1     int i = 0;
2     while (i < 5) {
3         puts("HELLO!");
4         ++i;
5     }
```

Стоит отметить, что условие ($i < 5$) будет проверено 6 раз и после завершения цикла, значение переменной i будет равно 5.

Переменная–счетчик.

Дано целое положительное n, вывести все числа меньше n.

```
1    int i = 0, n;
2    scanf("%d", &n);
3    while (i < n) {
4        printf("%d ", i);
5        i++;
6    }
```

Максимальный квадрат

Дано целое n от от 1 до 10000. Найти максимальный квадрат меньший n (то есть $x^2 < n$).

Ввод	30	100
Вывод	25	81

Решение

```
1  #include <stdio.h>
2
3  int main() {
4      int n, m = 1, ans;
5      scanf("%d", &n);
6      while (m * m < n) {
7          m++;
8      }
9      m--;
10     ans = m * m;
11     printf("%d\n", ans);
12     return 0;
13 }
```

Число цифр

Дано целое число от 1 до 10^{18}. Посчитать количество десятичных цифр.

Ввод	2017	12345678987654321
Вывод	4	17

Решение

```
1  #include <stdio.h>
2
3  int main() {
4      int ans = 0;
5      long long n;
6      scanf("%lld", &n);
7
8      while (n != 0) {
9          n /= 10;
10         ans++;
11     }
12     printf("%d\n", ans);
13     return 0;
14 }
```

Обратный порядок цифр

Дано целое число от 1 до 10^{18}. Вывести цифры числа в обратном порядке.

Ввод	2017	1000000
Вывод	7102	0000001

Решение

```
1  #include <stdio.h>
2
3  int main() {
4      int digit;
5      long long n;
6      scanf("%lld", &n);
7
8      while (n != 0) {
9          digit = n % 10;
10         printf("%d", digit);
11         n /= 10;
12     }
13     printf("\n");
14     return 0;
15 }
```

Делители числа

Дано целое n от от 1 до 30000. Вывести все делители числа n через пробел.

Ввод	25	12
Вывод	1 5 25	1 2 3 4 6 12

Решение

```
1  #include <stdio.h>
2
3  int main() {
4      int d = 1, n;
5      scanf("%d", &n);
6      while (d <= n) {
7          if (n % d == 0) {
8              printf("%d␣", d);
9          }
10         d++;
11     }
12     puts("");
13     return 0;
14 }
```

Простое ли число

Дано целое n от от 1 до 30000. Проверить, является ли данное число — простым (вывести `prime` или `not prime` соответственно).

Ввод	19	91
Вывод	`prime`	`not prime`

Решение

```
1  #include <stdio.h>
2
3  int main() {
4      int d = 2, tau = 0, n;
5      scanf("%d", &n);
6      while (d < n) {
7          if (n % d == 0) {
8              tau++;
9          }
10         d++;
11     }
12     if (tau == 0) {
13         printf("prime\n");
14     } else {
15         printf("not␣prime\n");
16     }
17     return 0;
18 }
```

Решение 2

```
1  #include <stdio.h>
2
3  int main() {
4      int d = 2, tau = 0, n;
5      scanf("%d", &n);
6      while (d * d <= n) {
7          if (n % d == 0) {
8              tau++;
9          }
10     }
11     if (tau == 0) {
12         printf("prime\n");
13     } else {
14         printf("not␣prime\n");
15     }
16     return 0;
17 }
```

Последовательная обработка чисел

Дана последовательность целых чисел от -1000 до 1000, причем ввод заканчивается нулем. Найти сумму чисел (гарантируется, что ответ по модулю не превосходит 10^9).

Ввод	1 2 -3 4 5 0	2 0
Вывод	9	2

Решение

```
1  #include <stdio.h>
2
3  int main() {
4      int a, sum = 0;
5      scanf("%d", &a);
6      while (a != 0) {
7          sum += a;
8          scanf("%d", &a);
9      }
10     printf("%d\n", sum);
11     return 0;
12 }
```

Минимум

Дана последовательность целых чисел от -1000 до 1000, причем ввод заканчивается нулем. Найти минимум чисел.

Ввод	2 3 1 4 5 0
Вывод	1

Решение

```
 1 #include <stdio.h>
 2
 3 int main() {
 4     int a, min;
 5     scanf("%d", &a);
 6     min = a;
 7     while (a != 0) {
 8         scanf("%d", &a);
 9         if (min > a) {
10             min = a;
11         }
12     }
13     printf("%d\n", min);
14     return 0;
15 }
```

Позиция минимума

Дано целое n от 1 до 1000. Далее n различных целых чисел от -1000 до 1000. Найти минимум из чисел и его позицию.

Ввод	5 2 3 1 4 5
Вывод	1 3

Решение

```
1  #include <stdio.h>
2
3  int main() {
4      int n, a, min, imin = 1, i = 1;
5      scanf("%d␣%d", &n, &min);
6      while (i < n) {
7          i++;
8          scanf("%d", &a);
9          if (a < min) {
10             min = a;
11             imin = i;
12         }
13     }
14     printf("%d␣%d\n", min, imin);
15     return 0;
16 }
```

Алгоритм Евклида

Даны два целых числа от 1 до 10^9. Найти наибольший общий делитель этих чисел.

Ввод	40 12	20 17
Вывод	4	1

Решение

$$(40, 12) = (12, 4) = (4, 0)$$

$$(a, b) = (b, a \bmod b)$$

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      int a, b, d;
6      scanf("%d␣%d", &a, &b);
7      while (b != 0) {
8          d = a % b;
9          a = b;
10         b = d;
11     }
12     printf("%d\n", a);
13     return 0;
14 }
```

Отладочная печать (простой метод)

Помогут дополнительные puts и printf.

```
1     int ans = 0, n = 100;
2     puts("1 step");
3     while (n % 10 == 0)
4         ans++;
5         n /= 10;
6     puts("2 step");
```

Обратите внимание, что printf() не гарантирует вывод на экран при зацикливании, если в вывод нет '\n', например, так

```
1     while (n % 10 == 0) {
2         printf("%d\n", n);
3         ans++;
4         n /= 10;
5     }
```

Отладка в gdb (сложный метод)

Компилируем

```
1    gcc prog.c -o prog -Wall -Werror -lm -g
```

Запускаем отладчик

```
1    gdb prog
```

Посмотрим код

```
1     list 1
```

Ставим точку останова (до которой программа будет выполняться в обычном режиме). Лучше ставить сразу после ввода.

```
1     break 6
```

Запускаем

```
1     run
```

Добавляем переменную наблюдения (можно несколько переменных)

```
1     display n
2     display ans
```

Делаем построчное выполнение (первый раз надо набрать команду целиком, потом просто Enter).

```
1     next
```

Выход

```
1     quit
```

1. Описать цикл, который печатает слово `Hello!` в цикле 100 раз.
2. Что будет выведено:

```
1     int n = 100, ans = 0;
2     while (n > 0){
3         n = n / 5;
4         printf("%d␣", n);
5     }
```

3. Что вычисляет данный код ($s = f(n)$)?

```
1     int n, s = 0, d = 2;
2     scanf("%d", &n)
3     while (d < n){
4         if (n % d == 0)
5             s += d;
6         d++;
7     }
```

4. Дано целое положительное число. Посчитать сумму цифр числа.
5. Дана последовательность положительных целых чисел. Ввод заканчивается нулем. Посчитать количество чётных чисел.