Особенности формата ввода-вывода

При реализации программ необходимо обратить внимание на следующее.

Во входных данных все числа вводятся ровно так, как в примерах, а именно, каждое число вводится в отдельной строке.

Программа не должна выводить никаких сообщений, кроме того, что требуется вывести в условии задачи (как правило, это одно или несколько целых чисел), в том числе нельзя выводить сообщения вида “Введите число”.

Целые числа во входных и выходных данных записываются только цифрами, то есть недопустимо использование записи 1000000.0 или 1e6 вместо числа 1000000.

Особенности отдельных языков программирования

Visual C++ — в тестирующей системе этот компилятор отсутствует, вместо него следует использовать GNU C++. В программах на Visual C++ не должны использоваться предварительно откомпилированные заголовочные файлы (директива препроцессора `#include "stdafx.h"`), не должно быть вызова `system("pause")`, иных нестандартных особенностей компилятора Visual C++. Программа должна быть написана в соответствии со стандартом языка C++.

Free Pascal — при сдаче решений на Free Pascal следует учитывать, что в используемых стандартных настройках компилятора (без использования специальных опций компиляции в тексте программы) тип integer является 16-битным, а длина строки ограничена 255 символами. Обратите внимание на то, что в тестирующей системе также доступен компилятор PascalABC.Net.

Примеры программ

Ниже приведены примеры программы, вычисляющей сумму двух целых чисел, считываемых со стандартного ввода и выводящей результат на стандартный вывод, с использованием 64-битных целых чисел. Входные числа записаны в двух разных строках.

Pascal

```
var a, b: int64;
begin
    read(a);
    read(b);
    writeln(a + b);
end.
```

Python версии 3

```
a = int(input())
b = int(input())
print(a + b)
```

C

```
#include <stdio.h>

int main()
{
  long long a, b;
  scanf("%lld%lld", &a, &b);
  printf("%lld\n", a + b);
  return 0;
}
```

C++

```
#include <iostream>

using namespace std;

int main()
{
  long long a, b;
  cin >> a >> b;
  cout << a + b << "\n";
  return 0;
}
```

C#

```
using System;
using System.IO;
class MainClass
{
  static void Main()
  {
    long a = long.Parse(Console.ReadLine());;
    long b = long.Parse(Console.ReadLine());
    Console.WriteLine("{0}", a + b);
  }
}
```

Java

В программах на Java не должно быть строки `package`.

```
import java.util.Scanner;

public class Main
{
  public static void main(String args[]) throws Exception
  {
    Scanner in = new Scanner(System.in);
    long a, b;
    a = in.nextLong();
    b = in.nextLong();
    System.out.println(a + b);
  }
}
```

Кумир (используются 32-битные числа)

```
алг Задача1
нач
  цел A, B
  ввод A
  ввод B
  вывод A + B
кон
```

QBasic (используется 32-битные числа)

```
DIM A AS LONG
DIM B AS LONG
INPUT A
INPUT B
PRINT A + B
```

Visual Basic

```
Module ProgramA
Sub Main()
DIM A, B AS Long
A = CLng(Console.Readline())
B = CLng(Console.Readline())
Console.WriteLine(A + B)
End Sub
End Module
```

PHP

```
<?php
$A = fgets(STDIN);
$B = fgets(STDIN);
print $A + $B;
?>
```

Perl

```
my $a, $b;
$a = <>;
$b = <>;
print $a + $b;
```

Ruby

```
a = gets.to_i
b = gets.to_i
puts a + b
```

Go

```
package main
import "fmt"
import "runtime/debug"

func main(){
    var a, b int64
    debug.SetMaxStack(16 * 1024 * 1024);
    fmt.Scan(&a)
    fmt.Scan(&b)
    fmt.Printf("%d\n", a + b)
}
```

Kotlin

```
import java.util.*

fun main(args: Array<String>) {
    val sc = Scanner(System.`in`);
    var a: Long = sc.next().toLong();
    var b: Long = sc.next().toLong();
    println(a + b);
}
```

Rust

```
use std::io;
fn main() {
    let mut a = String::new();
    io::stdin().read_line(&mut a).expect("");
    let a: i64 = a.trim().parse().expect("");
    let mut b = String::new();
    io::stdin().read_line(&mut b).expect("");
    let b: i64 = b.trim().parse().expect("");
    println!("{:?}", a + b);
}
```