Оформление задач

Во время тура участники решают предложенные задачи. Решенная задача представляется в виде файла в 8-битной кодировке (cp1251, cp866, koi8r, utf8) с исходным текстом программы на одном из языков программирования.

Решение не может содержать иных частей, таких как модули, пакеты (кроме стандартных) оформленных отдельно от основной программы. Все решения должны быть консольными приложениями.

В конце работы программа должна закрыть все открытые файлы (если таковые имеются) и завершиться с кодом возврата 0. Во всех языках программирования кроме Си и C++ нулевой код завершения программы вырабатывается автоматически при её нормальном завершении. В языках Си и C++ в конце функции main необходимо использовать оператор return 0.

Программа должна считывать входные данные со стандартного потока ввода (считывать с клавиатуры), результат работы должен выводиться на стандартный поток вывода (экран), если иное не оговорено условиями задачи.

Во всех языках программирования, перечисленных в п. 2.1. предусмотрены функции вывода данных на стандартный поток вывода и ввода данных со стандартного потока ввода.

В качестве примера оформления решений рассмотрено решение задачи нахождения суммы двух чисел (далее «A+B»).

Оформление решений на языке Си

В программах на языке Си для чтения данных следует использовать функции scanf, getchar и т. д., а для вывода результата — функции printf, putchar и т. д. Вместо типа данных __int64, поддерживаемого компилятором Microsoft C/C++, необходимо использовать тип данных long long.

В систему для проверки необходимо отправлять файл *.c. Ниже приведен пример решения задачи A+B:

```
#include <stdio.h>

int main(void)
{
  int a, b;
  scanf("%d%d", &a, &b);
  printf("%d\n", a + b);

  return 0;
}
```

Оформление решений на языке C++

В программах на языке C++ можно использовать как функции ввода-вывода языка Си, так и операции чтения из cin и записи в cout. Вместо типа данных __int64, поддерживаемого компилятором Microsoft C/C++, необходимо использовать тип данных long long.

В систему для проверки необходимо отправлять файл *.cpp. Ниже приведен пример решения задачи A+B:

```
#include <iostream>

using namespace std;

int main()
{
  int a, b;

  cin >> a;
  cin >> b;
  cout << a + b;

  return 0;
}
```

Во время выполнения допускаются исключения, возникающие внутри блока try..catch. Они будут проигнорированы системой. Исключения, возникающие вне блока try..catch приведут к Runtime Error.

Оформление решений на языке Free Pascal

В программах на языке Pascal для чтения следует использовать процедуры read или readln без файлового параметра, а для записи — процедуры write или writeln без файлового параметра.

В систему для проверки необходимо отправлять файл *.pas. Ниже приведены примеры решения задачи A+B:

```
program sum;

var a, b, c : longint;

begin
   readln(a);
   readln(b);

   c := a + b;
   writeln(c);
end.
```

Во время выполнения допускаются исключения, возникающие внутри блока try..except. Они будут проигнорированы системой. Исключения, возникающие вне блока try..except приведут к Runtime Error.

Оформление решений на языке Java

В программах на языке Java для чтения следует использовать методы работы с потоком System.in, а результат выводить в поток System.out. Для чтения данных можно использовать класс Scanner, но рекомендуется создавать StreamTokenizer.

На сервер для проверки необходимо отправлять файл *. java. Ниже приведен пример решения задачи A+B:

```
import java.io.*;
import java.util.Scanner;

public final class main
{
  public static void main(String args[]) throws Exception
  {
    Scanner s = new Scanner(System.in);
    int a, b;

    a = s.nextInt();
    b = s.nextInt();
    System.out.println(a + b);
  }
}
```

Оформление решений на языке VB.NET (компилятор mono Visual Basic)

В программах на языке VB.NET для чтения следует использовать методы работы с объектом Console: ReadLine, WriteLine. Следует учитывать, что система использует компилятор mono Visual Basic, который не в полной мере может поддерживать последние возможности языка Visual Basic .NET.

В систему для проверки необходимо отправлять файл *. vb. Ниже приведен пример решения задачи A+B:

```
Module ab
  Sub Main()
    Dim a, b, c as Integer
    a = Console.ReadLine()
    b = Console.ReadLine()
    c = a + b
    Console.WriteLine(c)
  End Sub
End Module
```

Особенности языков и сред программирования

Система и предоставляемый набор компиляторов работают под управлением операционной системы Linux, поэтому необходимо учитывать ряд особенностей приведенных ниже.

Компиляторы gcc и g++

В среде Linux в качестве разделителей каталогов в пути используется только символ «/», в отличие от Windows, где допускается как «/», так и «\». Поэтому в директивах #include должен использоваться только символ «/».

Неправильно:

```
#include <sys\types.h>
```

Правильно:

```
#include <sys/types.h>
```

Языки Си, C++

В среде Visual C++ для определения 64-битных целых типов используется тип __int64. Visual C++, начиная с версии 2005, поддерживает и стандартный 64-битный тип long long.

Обратите внимание, что компиляторы GCC, работающие на Windows (в среде Cygwin или MinGW), поддерживают только стандартный 64-битный тип long long.

Язык C++

Потоковый ввод и вывод языка C++ примерно в 10–20 раз медленнее функций printf и scanf. Если требуется ввести или вывести большое количество данных, используйте функции ввода-вывода языка Си.