

Список языков программирования и требования к программам на школьном этапе всероссийской олимпиады школьников по информатике на платформе «Сириус.Курсы» в 2023/24 учебном году

В задачах по программированию на проверку необходимо сдать текстовый файл, подготовленный в какой-либо среде разработки на компьютере. Файл должен содержать только текст программы и никакой служебной информации, например, XML-разметки Jupyter Notebook и т.д. Не рекомендуется использовать мобильные устройства (телефоны, планшеты) и онлайн-среды разработки (в том числе Jupyter Notebook). На школьном этапе всероссийской олимпиады по информатике тестирующая система будет поддерживать следующие языки программирования:

- Python 3
- C и C++
- Pascal
- Java
- C#
- Kotlin
- Go
- PHP
- Кумир
- Rust

Сохраните решение в простом текстовом файле (например, файл с расширением `cpp` для программы на C++, с расширением `py` для программы на Python и т.д.).

Решение должно в точности соответствовать условию задачи. В частности, программа должна считывать и выводить данные в том виде, в котором это описано в условии. Обратите внимание на следующее:

1. Во входных данных каждое число задано в отдельной строке, и вводить числа нужно по одному, нажимая «Enter» после каждого ввода.
2. Программа не должна выводить никаких иных сообщений, кроме того, что описано в условии задачи. В частности, нельзя выводить сообщения вида «Введите число», «Ответ» и т. д. Нельзя осуществлять какой-либо дополнительный отладочный вывод.
3. Целые числа во входных и выходных данных записываются только цифрами, то есть недопустимо использование записи `1000000.0` или `1e6` вместо числа `1000000`.

В случае возникновения каких-либо технических вопросов свяжитесь с нашей техподдержкой по адресу olymp@sochisirius.ru. Перед обращением рекомендуем обновить страницу олимпиады.

Примеры реализации программ на разных языках программирования

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.

Python 3

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

C++

```
#include <iostream>
using namespace std;
int main()
{
    long long a, b;
    cin >> a >> b;
    cout << a + b << "\n";
    return 0;
}
```

Pascal

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

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);
    }
}
```

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);
    }
}
```

Kotlin

```
import java.util.*

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

PHP

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

Кумир

```
алг Задача1
нач
    цел А, В
    ввод А
    ввод В
    вывод А + В
кон
```

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);
}
```
