### Всероссийская олимпиада школьников по информатике, 2017/18уч. год

### Школьный этап

Задания для 9–11 классов

Каждая задача оценивается в 100 баллов.

Ограничение по времени работы программы в каждой задаче – 1 секунда.

Задания принимаются на проверку и оцениваются, только если они выдают правильный ответ на всех примерах входных и выходных данных, приведённых в условии задачи. Программа не должна выводить никаких иных сообщений, кроме того, что требуется найти в задаче.

# Задача 1. Покупка

Ручка стоила *K* рублей. Первого сентября стоимость ручки увеличилась ровно на *P* процентов. Определите, сколько ручек можно купить на *S* рублей после подорожания.

Программа получает на вход три целых положительных числа. Первое число *K* – стоимость ручки в рублях до подорожания. Второе число *P –* величина подорожания ручки в процентах. Третье число *S* – имеющаяся сумма денег. Числа *K* и *S* не превосходят 107, число *P* не превосходит 100.

**Пример входных и выходных данных**

|  |  |  |
| --- | --- | --- |
| **Ввод** | **Вывод** | **Примечание** |
| 33 | 2 | Ручка стоила 33 рубля. После подорожания на 5 % ручка будет |
| 5  100 |  | стоить 34 рубля 65 копеек (заметим, что, поскольку первоначальная цена ручки была целым числом рублей, после подорожания |
|  |  | стоимость ручки будет выражаться целым числом рублей и копеек). |
|  |  | На 100 рублей после подорожания можно купить 2 ручки. |

***Система оценивания***

Решение, правильно работающее только для случаев, когда числа *K* и *S* не превосходят 100, будет оцениваться в 60 баллов.

***Примеры оформления решения задачи***

Ниже даны примеры ввода и вывода данных к этой задаче на нескольких языках программирования. Выберите один из языков программирования, допишите соответствующую программу и отправьте её на проверку с использованием одного из допустимых компиляторов.

|  |  |
| --- | --- |
| **Алгоритмический язык**  Интерпретатор Кумир | **Язык программирования Pascal**  Компиляторы Free Pascal, Borland Delphi, Pascal ABC.NET |
| алг Задача1 нач  цел k, p, s, result ввод k  ввод p ввод s  …  result := …  …  вывод result  кон | var k, p, s, result: longint; begin  readln(k); readln(p); readln(s);  …  result := …  … writeln(result);  end. |

|  |  |
| --- | --- |
| **Язык программирования Basic**  Компилятор Free Basic (аналог qbasic) | **Язык программирования Basic**  Компилятор Mono Visual Basic |
| DIM k AS LONG DIM p AS LONG DIM s AS LONG  DIM result AS LONG INPUT k  INPUT p INPUT s  …  result = …  …  print result | Module ProgramA Sub Main()  DIM k, p, s, result AS INTEGER k = Cint(Console.Readline())  p = Cint(Console.Readline()) s = Cint(Console.Readline())  …  result = …  … Console.WriteLine(CStr(result)) End Sub  End Module |
| **Язык программирования C**  Компилятор GNU C | **Язык программирования C++**  Компилятор GNU C++ |
| #include<stdio.h> int main() {  int k, p, s, result; scanf("%d%d%d", &k, &p, &s);  …  result = …  …  printf("%d", result); return 0;  } | #include<iostream> using namespace std; int main() {  int k, p, s, result; cin >> k >> p >> s;  …  result = …  …  cout << result; return 0;  } |
| **Язык программирования Python**  Пример для версий 2 и 3 языка Python | **Язык программирования C#**  Компилятор Mono C# |
| k = int(input()) p = int(input()) s = int(input())  …  result = …  … print(result) | using System; using System.IO; class Program {  static void Main() { int k, p, s, result;  k = int.Parse(Console.ReadLine()); p = int.Parse(Console.ReadLine()); s = int.Parse(Console.ReadLine());  …  result = …  …  Console.WriteLine("{0}", result);  }  } |
| **Язык программирования PHP**  Работает в режиме CLI (без web-сервера) | **Язык программирования Java** |
| <?php  $k = fgets(STDIN);  $p = fgets(STDIN);  $s = fgets(STDIN);  ...  $result = …  ...  print $result;  ?> | import java.io.\*; public class Main {  public static void main(String[] args) throws Exception {  DataInputStream in = new DataInputStream(System.in);  int k, p, s, result;  k = Integer.parseInt(in.readLine()); p = Integer.parseInt(in.readLine()); s = Integer.parseInt(in.readLine());  …  result = …  … System.out.println(result);  }  } |

# Задача 2. Плот

Посередине озера плавает плот, имеющий форму прямоугольника. Стороны плота направлены вдоль параллелей и меридианов. Введём систему координат, в которой ось *OX* направлена на восток, а ось *ОY* – на север. Пусть юго-западный угол плота имеет координаты (*x*1, *y*1), северо-восточный угол – координаты (*x*2, *y*2).

Пловец находится в точке с координатами (*x*, *y*). Определите, к какой стороне плота (северной, южной, западной или восточной) или к какому углу плота (северо-западному, северо-восточному, юго-западному, юго-восточному) пловцу нужно плыть, чтобы как можно скорее добраться до плота.

Программа получает на вход шесть чисел в следующем порядке: *x*1, *y*1 (координаты юго-западного угла плота), *x*2, *y*2 (координаты



*Y*

(*x*, *y*)

(*x*2, *y*2)

W

(*x*1, *y*1)

N

E

S

северо-восточного угла плота), *x*, *y* (координаты пловца). Все числа целые и по модулю не превосходят 100. Гарантируется, что *x*1 < *x*2, *y*1 < *y*2, *x* ≠ *x*1, *x* ≠ *x*2, *y* ≠ *y*1, *y* ≠ *y*2, координаты пловца находятся вне плота.

Если пловцу следует плыть к северной

стороне плота, программа должна вывести символ

«N», к южной – символ «S», к западной – символ *X*

«W», к восточной – символ «E». Если пловцу следует плыть к углу плота, нужно вывести одну из следующих строк: «NW», «NE», «SW», «SE».

**Пример входных и выходных данных**

|  |  |  |
| --- | --- | --- |
| **Ввод** | **Вывод** | **Примечание** |
| -1 | NW | Картинка выше соответствует этому примеру. |
| -2 |  |  |
| 5 |  |  |
| 3 |  |  |
| -4 |  |  |
| 6 |  |  |

***Система оценивания***

Решение, правильно работающее для случаев, когда ответом является одна из сторон плота «N», «S», «W», «E», будет оцениваться в 60 баллов.

Решение, правильно работающее для случаев, когда ответом является один из углов

«NW», «NE», «SW», «SE», будет оцениваться в 40 баллов.

# Задача 3. Пакуем чемоданы!

Алёна собирает вещи в отпуск. С собой в самолёт она может взять ручную кладь и багаж. Для ручной клади у Алёны есть рюкзак, а для багажа – огромный чемодан.

По правилам перевозки масса ручной клади не должна превосходить *S* кг, а багаж может быть любой массы (за сверхнормативный багаж Алёна готова доплатить). Разумеется, наиболее ценные вещи – ноутбук, фотоаппарат, документы и т. д. – Алёна хочет положить в ручную кладь.

Алёна разложила все свои вещи в порядке уменьшения их ценности и начинает складывать наиболее ценные вещи в рюкзак. Она действует следующим образом – берёт самый ценный предмет, и если его масса не превосходит *S*, то кладёт его в рюкзак, иначе кладёт его в чемодан. Затем она берёт следующий по ценности предмет, если его можно положить в рюкзак, то есть если его масса вместе с массой уже положенных в рюкзак вещей

не превосходит *S*, то кладёт его в рюкзак, иначе в чемодан, и таким же образом процесс продолжается для всех предметов в порядке убывания их ценности.

Определите вес рюкзака и чемодана после того, как Алёна сложит все вещи.

Первая строка входных данных содержит число *S* – максимально разрешённый вес рюкзака. Во второй строке входных данных записано число *N* – количество предметов. В следующих *N* строках даны массы предметов, сами предметы перечислены в порядке убывания ценности (сначала указана масса самого ценного предмета, затем масса второго по ценности предмета и т. д.). Все числа натуральные, число *S* не превосходит 2×109, сумма весов всех предметов также не превосходит 2×109. Значение *N* не превосходит 105.

Программа должна вывести два числа – вес рюкзака и вес чемодана (вес пустого рюкзака и чемодана не учитывается).

**Пример входных и выходных данных**

|  |  |  |
| --- | --- | --- |
| **Ввод** | **Вывод** | **Примечание** |
| 20 | 18 | Максимально возможная масса рюкзака 20 кг. Дано |
| 5  6  10  5  2  3 | 8 | 5 предметов весом 6, 10, 5, 2, 3.  Сначала предмет весом 6 кладётся в рюкзак, затем предмет весом 10 тоже кладётся в рюкзак. Предмет весом 5 нельзя положить в рюкзак, так как тогда вес рюкзака станет 21 кг, поэтому предмет весом 5 кладётся в чемодан. Затем предмет весом 2 кладётся |
|  |  | в рюкзак, а предмет весом 3 – в чемодан. Вес |
|  |  | рюкзака 6 + 10 + 2 = 18, вес чемодана 5 + 3 = 8. |

***Система оценивания***

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в 40 баллов.

# Задача 4. Туристический налог

Для пополнения бюджета в стране Авалон, известной своими горными туристическими маршрутами, ввели новый налог для туристов. Величина налога пропорциональна длине маршрута, но, поскольку маршрут проходит по горам и пройденное расстояние, зависящее от высоты спуска и подъёма, подсчитать сложно, налог считается без учёта высоты, то есть величина налога пропорциональна горизонтальному перемещению, совершённому туристической группой. Кроме того, в силу старинного обычая все туристические группы должны перемещаться по горам Авалона строго с запада на восток.

Турфирма хочет сэкономить на налоге, поэтому она хочет разработать туристический маршрут с минимальной величиной налога. При этом, поскольку маршрут является горным, он должен содержать подъём в гору и спуск с горы, то есть на маршруте должна быть точка, которая находится строго выше начала и конца маршрута.

Турфирма составила карту гор Авалона, содержащую информацию о высоте гор при передвижении с запада на восток. Высоты гор измерены в точках через равные расстояния. Найдите на данной карте гор Авалона туристический маршрут минимальной длины, удовлетворяющий условию наличия подъёма и спуска.

Первая строка входных данных содержит число *N* – количество точек на карте гор Авалона. Следующие *N* строк содержат информацию о высоте гор в данных *N* точках при движении с запада на восток. Все числа натуральные, не превосходящие 105.

Программа должна вывести два числа – номер точки начала маршрута и номер точки окончания маршрута. Точки нумеруются от 1 до *N*. Если маршрута, удовлетворяющего условиям, не существует, программа должна вывести одно число 0.

**Пример входных и выходных данных**

|  |  |  |
| --- | --- | --- |
| **Ввод** | **Вывод** | **Примечание** |
| 7 | 3 | Дано 7 точек с высотами 18, 10, 15, 20, 20, 10, 3. |
| 18  10  15  20 | 6 | Самый короткий маршрут, содержащий подъём и спуск, – это 15, 20, 20, 10. Он начинается в точке номер 3 и заканчивается в точке номер 6. |
| 20 |  |  |
| 10 |  |  |
| 3 |  |  |
| 3 | 0 | Высота гор монотонно убывает, поэтому искомого |
| 9  8 |  | маршрута не существует. |
| 5 |  |  |

***Система оценивания***

Решение, правильно работающее только для случаев, когда все входные числа не превосходят 100, будет оцениваться в 40 баллов.

# Задача 5. Делимость

Сегодня в школе на уроке математики проходят делимость. Чтобы продемонстрировать свойства делимости, учитель выписал на доске все целые числа от 1 до *N* в несколько групп, при этом если одно число делится на другое, то они обязательно оказались в разных группах. Например, если взять *N* = 10, то получится 4 группы.

Первая группа: 1.

Вторая группа: 2, 7, 9.

Третья группа: 3, 4, 10.

Четвёртая группа: 5, 6, 8.

Вы уже догадались, что, поскольку любое число делится на 1, одна группа всегда будет состоять только из числа 1, но в остальном подобное разбиение можно выполнить различными способами. От вас требуется определить минимальное число групп, на которое можно разбить все числа от 1 до *N* в соответствии с приведённым выше условием.

Программа получает на вход одно натуральное число *N*, не превосходящее 109, и должна вывести одно число – искомое минимальное количество групп.

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| **Ввод** | **Вывод** |
| 10 | 4 |

***Система оценивания***

Решение, правильно работающее только для случаев, когда *N* не превосходит 20, будет оцениваться в 20 баллов.

Решение, правильно работающее только для случаев, когда *N* не превосходит 103, будет оцениваться в 40 баллов.

Решение, правильно работающее только для случаев, когда *N* не превосходит 104, будет оцениваться в 60 баллов.