

Обязательно ознакомьтесь с памяткой участника.
Обратите внимание на следующие пункты:

1. Сегодня проходит первый тур олимпиады, завтра (18-го декабря) состоится второй тур. Все участники первого тура приглашаются поучаствовать во втором.
2. Используйте **только** ввод с клавиатуры и вывод на экран. Никакие файлы `input.txt/output.txt` или другие использовать нельзя.
3. Не используйте в Паскале модуль `crt` (и, следовательно, функцию `clrscr`).
4. Программа **должна заканчивать** работу после вывода выходных данных. Не используйте дополнительный `read/readln` или `system("pause")` в конце программы.
5. Не выводите на экран какую-либо лишнюю информацию, например: `writeln("Vvedite n:")`. Программа должна выводить на экран **только** то, что написано в секции **Формат выходных данных** условия задачи.
6. Не проверяйте входные данные на корректность. Жюри **гарантирует**, что они удовлетворяют всем требованиям из секции задачи **Формат входных данных** условия задачи.

Памятка участника олимпиады

Следуйте следующим правилам во время участия в олимпиаде:

1. Соревнование продлится **5 часов**.
2. Запрещается использовать любые электронные источники и носители информации, включая, но не ограничиваясь ими: портативные компьютеры, калькуляторы, радиотелефоны, пейджеры, дискеты, компакт-дискеты и прочее.
3. Запрещается общаться с другими участниками в любой форме.
4. Запрещается пользоваться Интернетом (за исключением доступа к тестирующей системе, см. пункт 13).
5. Решением является программа, написанная на одном из следующих языков программирования:
 - Pascal ABC — расширение `pas`
 - Free Pascal — расширение `pp`
 - C++ (Visual Studio 2010) — расширение `cpp`
 - C++ 14 (MinGW 6) — расширение `sxx`
 - Java 8 — расширение `java`, главный класс `Solution` в пакете по умолчанию
 - Python 3 — расширение `py`
 - C# (Visual Studio 2010, .NET 4.0) — расширение `cs`
6. На проверку отдается исходный текст программы, а не исполняемый файл. Программа должна полностью содержаться в одном файле, использование своих модулей и заголовочных файлов не допускается.
7. Программа должна читать входные данные только с клавиатуры и выводить результат на экран. Работать с файлами или какими-либо другими ресурсами компьютера запрещено.
8. Результаты работы программы проверяются автоматически, поэтому программа должна *точно соблюдать* формат вывода, указанный в условии. Не допускается вывод дополнительных сообщений. Ваша программа должна заканчивать свою работу после вывода ответа, а не «зависать» в ожидании ввода с клавиатуры.

9. Гарантируется, что входные данные будут соответствовать формату, указанному в условии. Все ограничения, заявленные в условии задачи, будут выполняться в тестах жюри. Проверять их в программе не имеет смысла. Ограничения по времени указаны для компьютера с тактовой частотой 3.2 ГГц.
10. Максимальный балл за каждую задачу указан в условии. Частичные решения будут оцениваться исходя из набора пройденных тестов.
11. Старайтесь использовать как минимум 32-х битные типы данных, если это позволяет ограничение по памяти (например, `longint` в Паскале).
12. Имя главного класса для программ на Java — `Solution` (см. пример программы на Java ниже).
13. Если вы участвуете через сайт <http://acm.sgu.ru/contest-region>, то вы самостоятельно отправляете решения в тестирующую систему. Логин и пароль для входа на сайт вы должны получить от преподавателя. Ознакомьтесь с инструкцией, размещенной на сайте.
14. В противном случае ваши решения будут собраны с вашего компьютера преподавателем по окончании работы. Для этого именуруйте файлы своих решений таким образом: `xxxxdd.ext`, где `xxxx` — это ваш четырехбуквенный идентификатор, `dd` — это номер задачи, а `ext` — это расширение вашего языка программирования. Например, имя файла для решения задачи 01 участника `abcd` на языке Pascal должно иметь вид `abcd01.pas`. Решения на Java следует именовать таким же образом, при компиляции вашей программы файл будет переименован в `Solution.java`. Поэтому главный класс должен называться `Solution` и располагаться в пакете по-умолчанию.
15. Результаты будут доступны поздно вечером на сайте <http://acm.sgu.ru/olimp>, и также будут распространены через вашего учителя информатики.

Примеры решения задачи A+B

Ниже приведены решения задачи, в которой требуется считать два числа и вывести их сумму.

- Pascal:

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

- C++:

```
#include <iostream>

using namespace std;

int main() {
  int a, b;
  cin >> a >> b;
  cout << a + b << endl;
  return 0;
}
```

- Java:

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

public class Solution {
  public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    PrintWriter writer = new PrintWriter(System.out);
    int a = scan.nextInt();
    int b = scan.nextInt();
    writer.print(a + b);
    writer.close();
  }
}
```

- Python 3:

```
(a,b)=input().split()
print (int(a) + int(b))
```

- C#:

```
using System;
class Solution {
    static void Main() {
        String [] input = Console.ReadLine().Split(' ');
        Console.WriteLine(Convert.ToInt32(input[0]) +
            Convert.ToInt32(input[1]));
    }
}
```

Задача 1. Компот (20 баллов)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У Николая есть a лимонов, b яблок и c груш. Он решил приготовить из них компот. По рецепту фрукты должны быть в соотношении $1 : 2 : 4$, то есть на каждый лимон в компоте должно быть ровно два яблока и ровно 4 груши. При этом фрукты нельзя измельчать, ломать или разрезать, поэтому в компот обязательно должны попасть целиком фрукты — лимоны, яблоки и груши.

Перед вами стоит задача определить максимальное суммарное количество лимонов, яблок и груш, из которых можно сварить компот по рецепту. Возможна ситуация, когда Николай не сможет для приготовления компота использовать ни одного фрукта, в таком случае выведите 0.

Формат входных данных

В первой строке следует целое положительное число a ($1 \leq a \leq 1000$) — количество лимонов, которые есть у Николая.

Во второй строке следует целое положительное число b ($1 \leq b \leq 1000$) — количество яблок, которые есть у Николая.

В третьей строке следует целое положительное число c ($1 \leq c \leq 1000$) — количество груш, которые есть у Николая.

Формат выходных данных

Выведите максимальное суммарное количество лимонов, яблок и груш, из которых можно сварить компот по рецепту.

Примеры

стандартный ввод	стандартный вывод
2 5 7	7
4 7 13	21
2 3 2	0

Замечание

В первом примере Николай может использовать для приготовления компота 1 лимон, 2 яблока и 4 груши, поэтому ответ $1 + 2 + 4 = 7$.

Во втором примере Николай может использовать для приготовления компота 3 лимона, 6 яблок и 12 груш, поэтому ответ $3 + 6 + 12 = 21$.

В третьей примере Николаю не хватит груш, чтобы приготовить сколько-нибудь компота, поэтому ответ 0.

Задача 2. Конфеты на Новый год (25 баллов)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У Деда Мороза есть n конфет, которые он хочет поскорее раздать детям на Новый год.

Перед вами стоит задача определить максимальное число детей, которые получат конфеты от Деда Мороза, если он хочет, чтобы каждый из детей, которым он даст конфеты, получил **различное** целое положительное количество конфет. Дед Мороз хочет раздать все n конфет, которые у него есть.

Формат входных данных

В единственной строке записано целое положительное число n ($1 \leq n \leq 1000$) — количество конфет у Деда Мороза.

Формат выходных данных

В первую строку выведите целое число k — максимальное число детей, которые получат конфеты.

Во второй строке выведите k **различных** целых чисел — сколько конфет получит каждый из k детей. Сумма выведенных k чисел должна быть равна n , так как Дед Мороз хочет раздать все конфеты.

Если решений несколько, выведите любое.

Примеры

стандартный ввод	стандартный вывод
5	2 2 3
9	3 3 5 1
2	1 2

Замечание

В первом примере Дед Мороз сможет дать конфеты максимум двум детям. Например, первый ребёнок может получить 2 конфеты, а другой — 3 конфеты.

Во втором примере Дед Мороз сможет дать конфеты максимум трём детям. Например, первый ребёнок может получить 3 конфеты, второй может получить 5 конфет, а третий 1 конфету.

В третьем примере Дед Мороз должен отдать все свои конфеты одному ребёнку.

Задача 3. Расшифровка (30 баллов)

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Поликарп без ума от шифрования, поэтому он пишет Свете сообщения в зашифрованном виде. *Медианной буквой* в слове он называет ту, которая находится в середине слова. Если слово четной длины, то медианная буква та, которая

стоит левее из двух средних букв. Медианная буква выделена подчеркиванием в следующих примерах: contest, info. Если слово состоит из одной буквы, то, по определению выше, она и будет его медианной буквой.

Поликарп шифрует каждое слово следующим образом — он выписывает медианную букву слова, затем удаляет эту букву из слова и повторяет процесс до тех пор, пока в слове есть хоть одна буква. Например, слово **volga** он зашифрует в слово **logva**.

Вам задано слово s , зашифрованное Поликарпом, а ваша задача — расшифровать его.

Формат входных данных

В первой строке следует целое положительное число n ($1 \leq n \leq 2000$) — длина зашифрованного слова.

Во второй строке следует строка s длины n , состоящая из строчных букв латинского алфавита — зашифрованное слово.

Формат выходных данных

Выведите слово, которое зашифровал Поликарп.

Примеры

стандартный ввод	стандартный вывод
5 logva	volga
2 no	no
4 abba	baba

Замечание

В первом примере Поликарп зашифровал слово **volga**. Сначала он выписал букву **l**, стоящую в позиции 3, после чего его слово приняло вид **voga**. После этого, Поликарп выписал букву **o**, стоящую в позиции 2, а его слово стало равным **vga**. Затем, Поликарп выписал букву **g**, стоящую во второй позиции, после этого слово изменилось на **va**. Потом он выписал букву **v**, а затем букву **a**. Таким образом, зашифрованное слово выглядит как **logva**.

Во втором примере Поликарп зашифровал слово **no**. Сначала он выписал букву **n**, слово стало равным **o**, и он выписал букву **o**. Таким образом, в этом примере, слово и его зашифровка выглядят одинаково.

В третьем примере Поликарп зашифровал слово `baba`. Сначала он выписал букву `a`, стоящую в позиции 2, после чего слово стало равным `bba`. Затем он выписал букву `b`, стоящую в позиции 2, а его слово приняло вид `ba`. После этого он выписал сначала букву `b`, стоящую в позиции 1, слово приняло вид `a`, и он выписал эту букву `a`. Таким образом, зашифрованное слово выглядит как `abba`.

Задача 4. Трамвай (40 баллов)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Трамвай в Берляндии ходит от точки 0 до точки s и обратно, преодолевая 1 метр за t_1 секунд в обоих направлениях. То есть трамвай ходит равномерно прямолинейно, мгновенно разворачиваясь в точках $x = 0$ и $x = s$.

В точке x_1 находится Игорь. Ему нужно попасть в точку x_2 . Игорь преодолевает 1 метр за t_2 секунд.

Перед вами стоит задача определить минимальное время, за которое Игорь может добраться из точки x_1 в точку x_2 , если известно, где находится трамвай и в каком направлении он едет в момент, когда Игорь пришел в точку x_1 .

Игорь может неограниченное количество раз заходить в трамвай. Точки, в которых Игорь может заходить и выходить из трамвая, не обязательно целочисленные. Считайте, что любая посадка/посадка происходит мгновенно. Игорь может произвольным образом перемещаться по прямой (но не быстрее, чем 1 метр за t_2 секунд) или даже стоять на месте.

Формат входных данных

В первой строке следуют три целых числа s , x_1 и x_2 ($2 \leq s \leq 1000$, $0 \leq x_1, x_2 \leq s$, $x_1 \neq x_2$) — максимальная координата точки, до которой доезжает трамвай, точка, в которую придёт Игорь и точка, в которую ему нужно попасть.

Во второй строке следуют два целых числа t_1 и t_2 ($1 \leq t_1, t_2 \leq 1000$) — время в секундах, за которое трамвай преодолевает 1 метр, и время в секундах, за которое Игорь преодолевает 1 метр.

В третьей строке следуют два целых числа p и d ($1 \leq p \leq s - 1$) — позиция трамвая в момент, когда Игорь пришел в точку x_1 , а также направление движения трамвая в этот же момент. Если $d = -1$, то трамвай движется в направлении от точки s в точку 0. Если $d = 1$, то трамвай движется в направлении от точки 0 в точку s .

Формат выходных данных

Выведите минимальное время в секундах, за которое Игорь может добраться из точки x_1 в точку x_2 .

Примеры

стандартный ввод	стандартный вывод
4 2 4 3 4 1 1	8
5 4 0 1 2 3 1	7

Замечание

В первом примере Игорю выгоднее пойти пешком и не ждать трамвай. Таким образом, ему нужно пройти 2 метра и он сможет это сделать суммарно за 8 секунд, так как он преодолевает 1 метр за 4 секунды.

Во втором примере Игорь может, например, пойти в сторону точки x_2 и дойти до точки 1 за 6 секунд (так как ему нужно пройти для этого 3 метра, а он преодолевает 1 метр за 2 секунды). В этот момент времени, трамвай также будет находиться в точке 1, поэтому Игорь может зайти в него, и доехать оставшийся 1 метр за 1 секунду. Таким образом, Игорь доберётся до точки x_2 суммарно за 7 секунд.

Задача 5. Зеленый и черный чай (40 баллов)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Иннокентий очень любит чай и сегодня хочет выпить ровно n чашек чая. Он был выпил и больше, но у него осталось ровно n пакетиков чая, a из которых — зеленый чай и b — черный.

Иннокентий не любит пить более, чем k раз подряд один и тот же чай. Перед вами стоит задача определить порядок заваривания пакетиков, при котором Иннокентий сможет выпить n чашек чая, при этом он не будет пить один и тот же чай более k раз подряд, либо сообщить, что это невозможно.

Формат входных данных

В первой строке следуют четыре целых числа n, k, a и b ($1 \leq k \leq n \leq 10^5$, $0 \leq a, b \leq n$) — количество чашек чая, которые хочет выпить Иннокентий, максимальное количество чашек одного и того же чая, которые Иннокентий может выпить подряд, количество пакетиков с зеленым чаем и количество пакетиков с черным чаем. Гарантируется, что $a + b = n$.

Формат выходных данных

Если выпить n чашек чая не удастся, выведите "NO" (без кавычек).

В противном случае, выведите строку длины n , состоящую из символов 'G' и 'B'. Если очередной символ равен 'G', то очередная кружка должна быть с зеленым чаем. Если очередной символ равен 'B', то очередная кружка должна быть с черным чаем. Если ответов несколько, разрешается вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
5 1 3 2	GBGBG
7 2 2 5	BGBGBB
4 3 4 0	NO

Задача 6. Замена чисел (45 баллов)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У Евгения есть n карточек, на каждой из которых написано ровно одно целое число. Евгений хочет поменяться с Николаем некоторыми карточками так, чтобы количество четных чисел на его карточках стало равно количеству нечетных чисел на его карточках, при этом все числа стали **различными**.

У Николая есть m карточек, на которых написаны различные числа от 1 до m , то есть у Николая есть ровно одна карточка, на которой написано число 1, ровно одна карточка, на которой написано число 2 и так далее.

Перед вами стоит задача найти минимальное количество обменов карт, а также определить какие на какие карточки нужно обменивать Евгению.

Формат входных данных

В первой строке следуют два целых числа n и m ($2 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq 10^9$) — количество карточек Евгения и количество карточек Николая. Гарантируется,

что n четно.

Во второй строке следует последовательность из n целых положительных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — номера на карточках Евгения.

Формат выходных данных

Если ответа не существует, выведите -1.

В противном случае, в первую строку выведите минимальное количество обменов. Во вторую строку выведите n чисел — карточки Евгения после обмена с Николаем. Порядок карточек должен совпадать с порядком карт из входных данных. Если i -я карточка не обменивалась, то число написанное на ней должно совпадать с числом из входных данных, в противном случае, будет считаться, что эта карточка была поменяна. Если ответов несколько, разрешается вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
6 2 5 6 7 9 4 5	1 5 6 7 9 4 2
8 6 7 7 7 7 8 8 8 8	6 7 2 4 6 8 1 3 5
4 1 4 2 1 10	-1

Задача 7. Музыка в машине (50 баллов)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Саша ездит на работу на машине, и дорога занимает ровно k минут. По дороге он слушает музыку. Все песни в его плейлисте расположены друг за другом и от прослушивания i -й из них Саша получает удовольствие равное a_i . При этом i -я песня длится t_i минут.

Перед началом пути Саша включит произвольную песню под номером x , и будет слушать песни по очереди: сначала песню под номером x , затем песню под номером $x+1$, затем песню под номером $x+2$, и так далее. Он будет слушать песни пока не придет на работу, либо пока не послушает последнюю песню из своего плейлиста.

Каждую песню Саша может либо прослушать полностью, либо прослушать ее *частично*. Во втором случае он слушает песню целое число минут хотя бы половину её длительности. Формально, если длина песни равна d минут, то Саша прослушает ее в течение не менее $\lceil \frac{d}{2} \rceil$ минут, а затем сразу переключит на следующую песню (если она есть). Например, если длина песни, которую Саша будет слушать *частично*, равна 5 минутам, то для получения удовольствия от прослушивания этой песни, он должен прослушать её хотя бы 3 минуты, а если длина песни равна 8 минутам — хотя бы 4 минуты. Переключение песен происходит мгновенно.

При этом Саша согласен прослушать *частично* не более w песен. Если последняя прослушанная песня играла менее, чем половину своей длительности, то она не приносит Саше удовольствия и не учитывается в песнях, прослушанных *частично*. Пропускать песни нельзя. Удовольствие от песни не зависит от режима ее прослушивания, для i -й песни эта величина равна a_i .

Помогите Саше выбрать такое x и выбрать не более w песен для *частичного* прослушивания, чтобы он получил максимальное удовольствие. Напишите программу находящую максимальное удовольствие, которое может получить Саша, от прослушивания песен по дороге на работу.

Формат входных данных

В первой строке следуют три целых числа n , w и k ($1 \leq w \leq n \leq 2 \cdot 10^5$, $1 \leq k \leq 2 \cdot 10^9$) — количество песен в плейлисте, количество песен, которые Саша согласен прослушать *частично* и время в минутах, за которое Саша доезжает до работы.

Во второй строке следуют n целых положительных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$), где a_i равно удовольствию, которое Саша получит от прослушивания i -й песни.

В третьей строке следуют n целых положительных чисел t_1, t_2, \dots, t_n ($2 \leq t_i \leq 10^4$), где t_i равно длительности i -й песни в минутах.

Формат выходных данных

Выведите максимальное удовольствие, которое может получить Саша, от прослушивания песен по дороге на работу.

Примеры

стандартный ввод	стандартный вывод
7 2 11 3 4 3 5 1 4 6 7 7 3 6 5 3 9	12
8 4 20 5 6 4 3 7 5 4 1 10 12 5 12 14 8 5 8	19
1 1 5 6 9	6
1 1 3 4 7	0

Замечание

В первом примере Саше нужно начать прослушивание с песни номер 2. Ее он должен прослушать *частично* (за 4 минуты), затем полностью прослушать песню номер 3 (за 3 минуты) и потом *частично* прослушать песню номер 4 (за 3 минуты). После прослушивания этих песен Саша получит удовольствие равное $4+3+5 = 12$. Песню номер 5 Саша прослушать не успеет, так как потратит на прослушивание песен 2, 3 и 4 время равное $4 + 3 + 3 = 10$ минутам и до работы останется ехать всего 1 минуту.

Задача 8. Новые дороги (50 баллов)

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Берляндии есть n городов, каждый из которых имеет свой уникальный номер — целое число от 1 до n , причём столица имеет номер 1. В Берляндии в настоящий момент совсем плохо с дорогами — их нет.

По этой причине было решено построить $n - 1$ дорогу так, чтобы между каждой парой городов существовал ровно один путь по дорогам.

В плане на постройку дорог было указано t чисел a_1, a_2, \dots, a_t , где t равно расстоянию от столицы до самого удаленного города, если добираться до него только

по новым дорогам, а a_i равно количеству городов, которые должны находиться на расстоянии i от столицы. Под расстоянием понимается количество дорог, которые нужно преодолеть на пути от одного города до другого.

Также было решено, что среди всех городов, исключая столицу, должно быть ровно k городов, к которым примыкает ровно одна дорога. Такие города являются тупиковыми и не могут быть экономически привлекательными. При подсчете таких городов столица не учитывается независимо от количества примыкающих дорог к ней.

Перед вами стоит задача предложить план постройки дорог, удовлетворяющий всем описанным условиям, либо сообщить, что это невозможно.

Формат входных данных

В первой строке следуют три целых положительных числа n , t и k ($2 \leq n \leq 2 \cdot 10^5$, $1 \leq t, k < n$) — расстояние до максимально удалённого города от столицы и количество городов, которые должны быть тупиковыми (столица в этом количестве не учитывается).

Во второй строке следует последовательность из t целых чисел a_1, a_2, \dots, a_t ($1 \leq a_i < n$), где i -е число равно количеству городов, которые должны находиться на расстоянии i от столицы. Гарантируется, что сумма всех значений a_i равна $n-1$.

Формат выходных данных

Если построить дороги, удовлетворяющие всем условиям, невозможно, выведите -1.

В противном случае, выведите в первую строку одно целое число n — количество городов в Берляндии. В следующей $n-1$ строке выведите по два целых числа — номера городов, которые соединяет очередная дорога. Каждая дорога должна быть выведена ровно один раз. Порядок вывода дорог и соединяемых дорогой городов неважен. Если решений несколько, выведите любое. Помните, что столица имеет номер 1.

Примеры

стандартный ввод	стандартный вывод
7 3 3 2 3 1	7 3 5 6 2 3 1 4 7 2 4 1 2
14 5 6 4 4 2 2 1	14 11 6 8 4 12 10 7 3 1 2 13 10 4 1 5 9 1 3 10 6 1 5 6 2 12 14
3 1 1 2	-1