

# НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*  
Школа *Състезателно програмиране*  
СЪСТЕЗАНИЕ, 9 март 2019 г.

---

## Задача А. Търсене на сума

Дадена е крайна числова редица. Да се намери дали дадено число може да се получи като сума от две члена на редицата.

*Вход.*

Стандартния вход съдържа няколко теста по два реда: на първия ред е числовата редица, на втория са числата за проверка.

*Ограничения.*

Дължината на редицата е от две до хиляда.

Числата в редицата са цели, неотрицателни и по-малки от хиляда.

*Изход.*

За всеки тестов пример на отделен ред да се отпечати редица от *yes/no* в зависимост от отговора на поредния тест.

Пример:

*Вход.*

8 2 4 3 3  
8 3 10 7  
10 20  
30

*Изход.*

no no yes yes  
yes

# НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*  
Школа *Състезателно програмиране*  
СЪСТЕЗАНИЕ, 9 март 2019 г.

---

## Задача В. Разстояние на Хеминг

Разстояние на Хеминг между два низа с еднаква дължина се дефинира като броя на различните съответни символи в единия и другия низ. Разстояние на Хеминг между две цели неотрицателни числа е разстоянието на Хеминг между низовете от единици и нули, представящи числата в двоична бройна система. За изравняване на дължините на низовете от единици и нули се добавят незначещи нули в началото на по-късия низ. В дадено множество от числа да се намери  $\max$  и  $\min$  разстояние между две (различни) числа от множеството.

*Вход.*

На стандартния вход са зададени няколко теста, всеки на отделен ред, съдържащ числата от множеството.

*Ограничения.*

Числата в множеството са в интервала  $[0, 2^{64}]$ .

Броят на числата в множеството е най-малко 2 и най-много 100.

*Изход.*

За всеки тестов пример на отделен ред да се отпечатаат получените минимум и максимум.

Пример:

*Вход.*

2 4 9 10

1 2 4 8 16 32

*Изход.*

1 3

2 2

# НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*  
Школа *Състезателно програмиране*  
СЪСТЕЗАНИЕ, 9 март 2019 г.

---

## Задача С. Интересни числа

Интересно число наричаме такова, на което поне половината от цифрите са еднакви. Евен-туални водещи нули не се броят! Например 3223 и 110 са интересни, докато 97791 и 123 не са.

Напишете програма, която намира броя на интересните числа между  $x$  и  $y$  включително.

*Вход.*

Всеки тестов пример е зададен на отделен ред на стандартния вход и се състои от естествените числата  $x$  и  $y$ , разделени с интервал.

*Ограничения.*

$$100 \leq x \leq y \leq 10^{18}$$

*Изход.*

За всеки тест извеждайте по една число на отделен ред -- броя на интересните числа между  $x$  и  $y$  включително.

Пример:

*Вход.*

110 133

*Изход.*

14

# НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*  
Школа *Състезателно програмиране*  
СЪСТЕЗАНИЕ, 9 март 2019 г.

---

## Задача D. Панграм

Панграм е изречение, включващо всички букви от дадена азбука.

Напишете програма, която да разпознава панграми. В конкретната задача азбуката е латинската. Едно изречение е панграм, ако съдържа поне веднъж всяка от 26-те букви, без значение дали е главна или малка буква.

*Вход.*

На първия ред на стандартния вход е зададен броя на тестовите примери  $n$ . Следват  $n$  реда. На всеки от тях ще има последователност от най-много 100 символа, задаващи поредния тест.

*Изход.*

Отговора за всеки тест извеждайте на отделен ред на стандартния изход. Ако дадено изречение е панграм, извеждайте `pangram`. В противен случай извеждайте `missing`, интервал и липсващите букви. Всички липсващи букви трябва да се малки и между тях не трябва да има интервали. Също така трябва да се подредени лексикографски.

Пример:

*Вход.*

```
3
The quick brown fox jumps over the lazy dog.
ZYXW, vu TSR Ponm lkj ihgfd CBA.
.,?!'" 92384 abcde FGHIJ
```

*Изход.*

```
pangram
missing eq
missing klmnopqrstuvwxyz
```

# НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*  
Школа *Състезателно програмиране*  
СЪСТЕЗАНИЕ, 9 март 2019 г.

---

## Задача Е. Думи

На Янко му попаднали две непразни думи, състоящи се от най-много 100000 малки латински букви. Той взема непразен префикс от първата дума и непразен суфикс от втората и ги конкатенира. Например от “tree” и “heap”, той може да получи думи като “treap”, “tap” или “theap”.

Задачата ви е, по зададени две думи да определите броя на различните думи, които могат да бъдат получени по описания начин.

*Вход.*

На първия ред на стандартния вход е зададен броя на тестовите примери  $n$ . Следват  $2n$  реда. Всеки тест се състои от два последователни реда, на които са зададени по една от думите.

*Изход.*

Отговора за всеки тест извеждайте на отделен ред на стандартния изход.

Пример:

<i>Вход.</i>	<i>Изход.</i>
2	9
cat	14
dog	
tree	
heap	

# НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*  
Школа *Състезателно програмиране*  
СЪСТЕЗАНИЕ, 9 март 2019 г.

---

## Задача F. Двоични дървета за търсене

Двоичните дървета за търсене (Binary Search Tree) са много популярна и ефективна структура от данни. При тях всички стойности в лявото на корена поддърво са по-малки от стойността на корена, както и всички стойности в дясното на корена поддърво са по-малки от стойността на корена. Също така всяко поддърво е двоично дърво за търсене. В конкретната задача не се интересуваме от празните дървета. В тази задача стойностите на възлите ще са цели положителни числа, не по-големи от  $10^9$ .

Обхождане на дървото *корен-ляво-дясно* (*preorder traversal*) може да се дефинира, чрез псевдокод по следния начин:

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```

Напишете програма, която определя дали дадена последователност от числа е такова обхождане на двоично дърво за търсене.

*Вход.*

Всеки тестов пример е зададен на отделен ред на стандартния вход и се състои от не повече от 1000 числа.

*Изход.*

За всеки тестов пример извеждайте на нов ред на стандартния изход "Case x: ", където x е номера на поредния тестов пример, последвано от "yes" или "no", без кавичките, в зависимост от това, дали числата от тестовия пример са валидно обхождане *корен-ляво-дясно*.

Пример:

*Вход.*

2 3 1  
50 30 20 10 25 40 45 70 90 80

*Изход.*

Case 1: no  
Case 2: yes

# НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*  
Школа *Състезателно програмиране*  
СЪСТЕЗАНИЕ, 9 март 2019 г.

---

## Задача G. Формула

Всеки знае редицата на Фибоначи:

$$f_1 = f_2 = 1; f_{n+1} = f_n + f_{n-1}$$

и не е много трудно да разгадае формулата, ако разполагаме само с няколко члена на редицата, напр.

1, 1, 2, 3, 5, 8, 13, 21.

Намерете формулата, по която е построена следната редица:

1, 5, 12, 22, 35, 51, 70, 92.

Напишете програма, която да намира  $n$ -тия член на редицата.

*Вход.*

На стандартния вход се задават числа  $n_1, n_2, \dots$

*Ограничения.*

$$1 \leq n_i \leq 10^8$$

*Изход.*

За всяко число  $n_i$  от входа на стандартния изход да се изведе  $n_i$ -тия член на редицата.

Пример:

*Вход.*

2 10 25

*Изход.*

5 145 925

# НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*  
Школа *Състезателно програмиране*  
СЪСТЕЗАНИЕ, 9 март 2019 г.

---

## Задача Н. Ветроходец

Ветроходец попаднал близо до остров в област на слаби, но постоянни ветрове, които сменяли посоката си с настъпване на деня и нощта. През деня духали към острова, а през нощта – в обратана посока. Всяка сутрин ветроходецът вдигал платната, а през нощта ги свалял. Така всеки ден успявал да се доближи с  $a$  мили до острова. За жалост през нощта обратно течение го връщало с  $b$  мили навътре в морето. Една сутрин GPS системата му отчетела с мили разстоянието до острова. След колко дни ветроходецът ще достигне острова?

*Вход.*

На стандартния вход за всеки тестов пример на един ред се задват числата  $a$ ,  $b$  и  $c$ .

*Ограничения.*

$$0 \leq a \leq 100$$

$$0 \leq b \leq 100$$

$$0 \leq c \leq 100$$

*Изход.*

За всеки тестов пример на отделен ред да се изведе 0, ако ветроходецът ще достигне острова същия ден; 1, ако ще достигне острова на следващия ден и т.н. Да се изведе  $-1$ , ако при дадените входни данни ветроходецът никога няма да достигне острова.

Пример:

*Вход.*

*Изход.*

10 5 8

0

10 5 14

1

10 11 12

-1