



Пред вама је 6 задатака чији је преглед дат у табели на следећој страни. **Ученици Б категорије решавају само прва 3 задатка, а ученици А категорије само последња 3 задатка.** Уколико сте ученик Б категорије а желите да пређете у А категорију, неопходно је да то наведете у коментарима кода који предајете и да о томе обавестите дежурног професора пре завршетка такмичења. Задатке морате радити самостално без коришћења помоћних материјала (литература, текстови на интернету, изворни кодови, ...). Израда задатака траје 5 сати.

Такмичење се реализује преко Петља система за такмичења. Потребно је да се на адреси **drzavno.petlja.org** улогујете у тај систем користећи своје корисничко име које сте користили за квалификације и Окружно такмичење.

Питања у вези са формулацијама задатака можете упутити преко Петља система за такмичење.

## ПОДЗАДАЦИ

- Тест примери за сваки задатак су подељени у групе које се називају подзадаци. Унутар сваког подзадатка важе посебна ограничења, која су наведена у тексту задатка.
- Да бисте добили бодове предвиђене за неки подзадатак, потребно је да исправно решите **све** тест примере из одговарајуће групе. Убрзо након слања задатка, задатак ће бити аутоматски прегледан и имаћете увид у то за које сте све подздатке освојили бодове. **У коначне бодове вам улази само резултат последњег послатог програма за сваки задатак.**
- Програм који пошаљете на тестирање ће бити тестиран и на примерима који су дати у тексту задатка. У приказу резултата то ћете видети као нулти подзадатак, који носи 0 бодова.

## НАПОМЕНЕ

- Подаци се читају/исписују преко стандардног улаза/излаза - **не користити фајлове!**
- Излазни подаци морају бити **тачно** у облику датом у опису задатка. **Немојте исписивати додатне ствари попут "Тражени број је...", "Унесите бројеве...", празне линије на почетку и сл.** Улазне податке учитавати **искључиво** у формату у коме су задати (**водити рачуна да ли су подаци у једној или у више линија**).
- На крају програма **обавезно уклонити** `"readln;"` и `"system('pause');"` наредбе!
- Уколико је потребно користити **64-битне бројеве** користите `int64` у Pascal-у, односно `long long` у C++-у; обратите пажњу да `long` у C++-у не мора увек бити 64-битни тип. За учитавање и испис 64-битних бројева у C++-у користити спецификатор `%lld`.
- Visual Studio, Lazarus, ABCPascal **нису званична окружења и не користе званичне компајлере**. Може вам се десити да се ваш код компајлира на овим компајлерима али не и на званичним (или да се другачије понаша). Користите CodeBlocks и Free Pascal.
- C++: Функција `main` мора бити декларисана као `"int main()"`. Експлицитно `include`-овати све библиотеке које користите (нпр. `<cstring>`, `<algorithm>`). Учитавање и испис са `cin` и `cout` је споро у случају великог броја података. Користити `scanf` и `printf`, или на почетку `main` функције додати `ios::sync_with_stdio(false);` и `cin.tie(0);`, у ком случају не можете користити `scanf` и `printf` у истом програму.

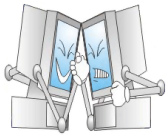


## Преглед задатака

2 / 15

Б категорија			
Задатак	Хакерисање	Пољопривредник	Минирање
Улаз	Стандардни улаз (stdin)		
Излаз	Стандардни излаз (stdout)		
Временско ограничење	1 s	0.5 s	2 s
Меморијско ограничење	64 MB	64 MB	128 MB
Број поена	100	100	100

А категорија			
Задатак	Нз	Визаард	Париз
Улаз	Стандардни улаз (stdin)		
Излаз	Стандардни излаз (stdout)		
Временско ограничење	1 s	1.5 s	1.5 s
Меморијско ограничење	512 MB	256 MB	256 MB
Број поена	100	100	100

**Б1. Хакерисање**

3 / 15

Млади хакер Жак је одлучио да по сваку цену дође до шифре налога свога професора. Открио је да је та шифра тачно једнака одговору на  $Q$  упита над професоровим низом. Професоров низ  $A_i$  има  $N$  елемената, а упити су типа  $L\ R$ . Одговор на упит је вредност битовске конјункције елемената низа између индекса  $L$  и  $R$  (укључујући и њих). Помозите Жаку да открије шифру.

**Опис улаза**

У првом реду налазе се бројеви  $N$ , дужина низа и  $Q$  највећи број упита над низом. У другом реду налази се низ од  $N$  елемената. У наредних  $Q$  редова налазе се упити описаног формата.

**Опис излаза**

За сваки од  $Q$  упита исписати вредност битовске конјункције елемената између индекса  $L$  и  $R$ .

**Пример****Улаз**

```
5 4
1 5 4 3 2
1 3
4 5
2 5
2 3
```

**Излаз**

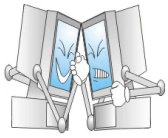
```
0
2
0
4
```

**Објашњење примера**

Одговор на први упит је:  $1 \text{ and } 5 \text{ and } 4 = 0$ . Одговор на други упит је:  $3 \text{ and } 2 = 2$ . Одговор на трећи упит је:  $5 \text{ and } 4 \text{ and } 3 \text{ and } 2 = 0$ . Одговор на четврти упит је:  $5 \text{ and } 4 = 4$ .

**Ограничења**

- $1 \leq Q, N \leq 2 \cdot 10^5$
- $0 \leq A_i \leq 10^6$

**Б1. Хакерисање**

4 / 15

Постоје три подзадатака:

- Подзадатак 1 [15 поена]:  $N, Q \leq 10^3$ .
- Подзадатак 2 [33 поена]:  $A_i \leq 1$ .
- Подзадатак 3 [52 поена]: Нема додатних ограничења.

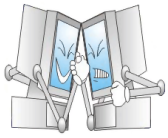
## Напомена

Оператор конјункције у Pascal-у је означен са `and`, док у C++ га записујемо помоћу симбола `&`. Ова операција  $x$  and  $y$  се за ненегативне целе бројеве  $x, y$  дефинише на следећи начин. Прво се бројеви запишу у бинарном запису. Уколико један број има мање цифара од другог, дописују му се водеће нуле све док не буду имали исти број бинарних цифара. Тако се добијају два низа бинарних цифара, означимо их са  $a_1, \dots, a_k$  и  $b_1, \dots, b_k$ . Затим се за сваку позицију  $i \in \{1, \dots, k\}$  рачуна  $c_i$  помоћу следећих правила:

- За  $a_i = 0, b_i = 0$  важи  $c_i = 0$
- За  $a_i = 0, b_i = 1$  важи  $c_i = 0$
- За  $a_i = 1, b_i = 0$  важи  $c_i = 0$
- За  $a_i = 1, b_i = 1$  важи  $c_i = 1$

Низ бинарних цифара  $c_1, \dots, c_k$  (који можда има водеће нуле) је бинарни запис резултата, односно броја  $x$  and  $y$ .

Битовска конјункција између  $n$  елемената  $x_1, x_2, \dots, x_n$  дефинише се као  $x_1$  and  $x_2$  and  $\dots$  and  $x_n = (\dots((x_1$  and  $x_2)$  and  $x_3)$  and  $x_4)\dots)$  and  $x_n$ .

**Б2. Пољопривредник**

5 / 15

Милоранка, нова млада нада Шљивогограда, тек што је научила да учитава низове, а већ је добила задатак од свог професора Растислава. Задатак гласи:

Дата су три броја  $N$ ,  $K$  и  $X$ . Пронаћи један низ  $A$  који има  $N$  чланова, сви чланови су природни бројеви мањи од  $10^6$  и у којем постоји тачно  $K$  парова индекса  $(i, j)$  са својством:

- $1 \leq i < j \leq N$
- $|A_i - A_j| \geq X$ .

Пошто је Растислав пољопривредник и бави се програмирањем једино када није сезона шљива, могуће да је погрешно и дао таква три броја да не постоји ниједан низ  $A$  који испуњава тражене услове. У том случају исписати  $-1$ . Помозите Милоранки да реши професоров задатак.

**Опис улаза**

У првој и јединој линији стандардног улаза налазе се три цела броја  $N$ ,  $K$  и  $X$ , описана у тексту задатка.

**Опис излаза**

У јединој линији стандардног излаза исписати  $N$  природних бројева, чланове низа  $A$  који испуњава тражена својства. У случају да такав низ не постоји исписати  $-1$ .

**Пример 1****Улаз**

```
3 2 5
```

**Излаз**

```
1 8 2
```

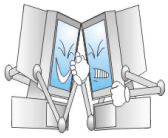
**Објашњење примера**

Низ  $A = 1, 8, 2$ , има  $N = 3$  члана, и  $K = 2$  пара индекса,  $(1, 2)$  и  $(2, 3)$ , испуњавају својство из задатка:

- $|A_1 - A_2| = 7 \geq 5 = X$
- $|A_2 - A_3| = 6 \geq 5 = X$

**Ограничења**

- $1 \leq N \leq 1000$
- $2 \leq X \leq 1000$
- $0 \leq K \leq \frac{N \cdot (N-1)}{2}$



## Б2. Пољопривредник

Постоје три подзадатака:

- Подзадатак 1 [21 поен]:  $K = 3$ .
- Подзадатак 2 [22 поена]:  $N = K$ .
- Подзадатак 3 [57 поена]: Нема додатних ограничења.

**Б3. Минирање**

7 / 15

Мали Данило треба да се нађе са својим најбољим другом, али због силних радова у граду то му представља проблем. Срећом, Данило увек носи пар мина са собом које може да искористи да стигне до свог друга.

Стање града се може описати матрицом, где је свако поље слободно или блокирано радовима. У сваком тренутку Данило може да пређе са поља на ком се налази на суседна слободна поља, где су суседна поља у матрици она поља која су непосредно изнад, испод, лево или десно од тренутног поља.

Данило сме да постави мину само на поље које му је достижно. Након експлозије прве мине, под условом да му је остало још мина, Данило може да постави следећу мину на сва поља до која су му тада достижна (могуће је да се скуп достижних поља проширио након експлозије прве мине).

Али Данило је заборавио снагу својих мина. Помозите му да нађе минималну снагу мина  $K$  која му је потребна да би стигао до свог друга.

Снага мина одређује величину њене експлозије. Мина снаге  $K$  ослобађа сва поља у правоугаонику величине  $(2K + 1) * (2K + 1)$ , са центром у истом пољу где је мина постављена.

На пример, ако се постави мина у средини леве матрице снаге  $K = 1$ , поља која је експлозија обухватила су означена са **0** и та поља ће на даље бити проходна.

X . . . .	X . . . .
XXXXX	X000X
X . . X .	----> X000 .
. . . X .	. 000 .
XX . . X	XX . . X

Нађи минималну потребну снагу мина,  $K$ , да би било могуће да Данило постави мине на неки начин тако да стигне до свог друга.

**Опис улаза**

У првом реду налазе се два броја  $N$ , који представља дужине странице матрице, и  $M$ , број мина који је Данило понео са собом. Наредних  $N$  редова представља стање града.

Слободна поља су означена карактером **0**, а блокирана поља са **X**. Данило се налази на пољу означеним са **A**, а његов друг на пољу означеним са **B**. Та два поља су слободна.

**Опис излаза**

Исписати природан број  $K$  који представља минималну јачину мина потребну да Данило стигне до његовог друга. Ако није потребно користити мине уопште, исписати -1.

**Пример 1**

**Б3. Минарање**

8 / 15

**Улаз**

```
6 1
A.XX..
..XX..
XXXX..
..XX..
..XX..
..XX..
..XX.B
```

**Излаз**

```
2
```

**Пример 2****Улаз**

```
7 2
AXXXXXX
XXXXXXX
XXXXXXX
XXXXXXX
XXXXXXX
XXXXXXX
XXXXXXX
XXXXXXB
```

**Излаз**

```
3
```

**Објашњење примера**

У првом примеру, Данило може да постави мину на поље (2, 2) или (2, 1) јачине  $K = 2$  да би направио пут до свог друга.

У другом примеру, Данило може само да стави мину на своју позицију. Стање матрице након прве експлозије мином снаге  $K = 3$  на поље (1, 1):





## Б3. Минирање

9 / 15

```
....XXX
....XXX
....XXX
....XXX
XXXXXXX
XXXXXXX
XXXXXXB
```

Сад Данило једино може да постави своју последњу мину на поље  $(4, 4)$  да би отворио пут до свог друга.

### Ограничења

- $2 \leq N \leq 500$
- $1 \leq M \leq 2$

Постоје три подзадатака:

- Подзадатак 1 [31 поена]:  $M = 1$ .
- Подзадатак 2 [32 поена]:  $N \leq 50$ .
- Подзадатак 3 [37 поена]: Нема додатних ограничења.

### Напомена

Данило и његов друг су отпорни на експлозије.



Низ који настане када се из неког низа обришу неки елементи се зове нз. За дати низ  $A$  и број  $K$  наћи лексикографско најмањи нз који се добија брисањем тачно  $K$  елемената низа  $A$ .

## Опис улаза

У првој линији налазе се два цела броја. Први број,  $N$ , представља дужину низа  $A$ , а други број је број  $K$ . У наредној линији се налази  $N$  целих бројева који представљају елементе низа  $A$ .

## Опис излаза

Исписати  $N - K$  целих бројева који представљају тражени нз.

## Пример 1

### Улаз

```
4 1
4 8 6 3
```

### Излаз

```
4 6 3
```

## Објашњење примера

Брисањем једног елемента могу се добити нзови `8 6 3`, `4 6 3`, `4 8 3` и `4 8 6`. Нз `4 6 3` је лексикографски мањи од свих осталих.

## Пример 2

### Улаз

```
8 3
6 0 3 1 4 2 1 7
```

### Излаз

```
0 1 2 1 7
```

## Ограничења



## А1. Нз

11 / 15

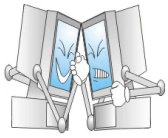
- $0 \leq A_i \leq 10^9$
- $0 \leq K \leq N - 1$

Постоји пет подзадатака:

- Подзадатак 1 [19 поена]:  $N \leq 400$  и  $K = 1$
- Подзадатак 2 [24 поена]:  $N \leq 200.000$  и  $A_i \in \{0, 1\}$
- Подзадатак 3 [18 поена]:  $N \leq 400$
- Подзадатак 4 [18 поена]:  $N \leq 200.000$  и  $A_i \leq 100$
- Подзадатак 5 [21 поена]:  $N \leq 500.000$

## Напомена

Кажемо да је низ  $P$  лексикографски мањи од низа  $Q$  исте дужине ако важи да је  $P_i < Q_i$  за најмање  $i$  за које је  $P_i \neq Q_i$ .



## А2. Визаард

12 / 15

Одавно се зна за мистична својства низова којима је и битовска конјункција свих елемената већа од нуле и битовска ексклузивна дисјункција већа од нуле. Чаробњак Визаард је добио један овакав низ са  $N$  елемената. Да би он добио мистична својства, Визаард на сваком од његових елемената може да примени једну од следеће две операције:

- Уколико је  $A_i > 0$ , може да смањи вредност елементу са индексом  $i$ . Овом операцијом вредност  $i$ -тог елемента се претвара у  $A_i - 1$ .
- Може да повећа вредност елементу са индексом  $i$ . Овом операцијом вредност  $i$ -тог елемента се претвара у  $A_i + 1$ .

Визаарда интересује колико најмање операција мора да примени тако да низ добије мистична својства, тј. да му битовска конјункција свих елемената буде већа од нуле и да му битовска ексклузивна дисјункција свих елемената буде већа од нуле.

### Опис улаза

У првој линији налази се један број  $N$ , дужина низа. У наредној линији налази се низ од  $N$  елемената.

### Опис излаза

Исписати најмањи број операција потребних да и битовска конјункција и битовска ексклузивна дисјункција свих елемената буде већа од 0.

### Пример

#### Улаз

```
4
5 2 3 3
```

#### Излаз

```
1
```

### Објашњење примера

Приметимо да је  $5 \text{ and } 2 \text{ and } 3 \text{ and } 3 = 0$ . Међутим приметимо да уколико други елемент смањимо за један, добијамо низ  $[5, 1, 3, 3]$ , за који важи  $5 \text{ and } 1 \text{ and } 3 \text{ and } 3 = 1$  и  $5 \text{ xor } 1 \text{ xor } 3 \text{ xor } 3 = 4$ .

### Ограничења

- $2 \leq N \leq 5 \cdot 10^5$

**A2. Визаард**

- $0 \leq A_i \leq 10^{18}$

Постоје четири подзадатака:

- Подзадатак 1 [14 поена]:  $N, A_i \leq 16$ .
- Подзадатак 2 [20 поена]:  $N \leq 10^3, A_i \leq 10^9$ .
- Подзадатак 3 [29 поена]:  $N \leq 3 \cdot 10^4, A_i \leq 10^9$ .
- Подзадатак 4 [37 поена]: Нема додатних ограничења.

## Напомена

Оператор конјункције у Pascal-у је означен са `and`, док у C++ га записујемо помоћу симбола `&`. Ова операција  $x$  and  $y$  се за ненегативне целе бројеве  $x, y$  дефинише на следећи начин. Прво се бројеви запишу у бинарном запису. Уколико један број има мање цифара од другог, дописују му се водеће нуле све док не буду имали исти број бинарних цифара. Тако се добијају два низа бинарних цифара, означимо их са  $a_1, \dots, a_k$  и  $b_1, \dots, b_k$ . Затим се за сваку позицију  $i \in \{1, \dots, k\}$  рачуна  $c_i$  помоћу следећих правила:

- За  $a_i = 0, b_i = 0$  важи  $c_i = 0$
- За  $a_i = 0, b_i = 1$  важи  $c_i = 0$
- За  $a_i = 1, b_i = 0$  важи  $c_i = 0$
- За  $a_i = 1, b_i = 1$  важи  $c_i = 1$

Низ бинарних цифара  $c_1, \dots, c_k$  (који можда има водеће нуле) је бинарни запис резултата, односно броја  $x$  and  $y$ .

Битовска конјункција између  $n$  елемената  $x_1, x_2, \dots, x_n$  дефинише се као  $x_1$  and  $x_2$  and  $\dots$  and  $x_n = (\dots((x_1$  and  $x_2)$  and  $x_3)$  and  $x_4)\dots)$  and  $x_n$ .

Оператор ексклузивне дисјункције у Pascal-у је означен са `xor`, док у C++ га записујемо помоћу симбола `^`. Ова операција  $x$  xor  $y$  се за ненегативне целе бројеве  $x, y$  дефинише на следећи начин. Прво се бројеви запишу у бинарном запису. Уколико један број има мање цифара од другог, дописују му се водеће нуле све док не буду имали исти број бинарних цифара. Тако се добијају два низа бинарних цифара, означимо их са  $a_1, \dots, a_k$  и  $b_1, \dots, b_k$ . Затим се за сваку позицију  $i \in \{1, \dots, k\}$  рачуна  $c_i$  помоћу следећих правила:

- За  $a_i = 0, b_i = 0$  важи  $c_i = 0$
- За  $a_i = 0, b_i = 1$  важи  $c_i = 1$
- За  $a_i = 1, b_i = 0$  важи  $c_i = 1$
- За  $a_i = 1, b_i = 1$  важи  $c_i = 0$

Низ бинарних цифара  $c_1, \dots, c_k$  (који можда има водеће нуле) је бинарни запис резултата, односно броја  $x$  xor  $y$ .

Битовска ексклузивна дисјункција између  $n$  елемената  $x_1, x_2, \dots, x_n$  дефинише се као  $x_1$  xor  $x_2$  xor  $\dots$  xor  $x_n = (\dots((x_1$  xor  $x_2)$  xor  $x_3)$  xor  $x_4)\dots)$  xor  $x_n$ .



Париз је познат по бројним атракцијама и знаменитостима које се могу видети у том граду. У последњој предизборној кампањи, градски оци Париза су изградили нове, лепе и модерне, пешачке путеве између атракција. Како би даље финансирани своју кампању, смислили су и начин како могу преварити несмотрене грађане и туристе - **пешачке путеве су направили једносмерним**, и свакоме ко се креће у супротном смеру биће написана казна. За сваку атракцију у граду, направљен је **тачно један пешачки пут који води до ње**.

Ана ће ове године први пут ићи у Париз, и са свиме овиме је добро упозната. Она јако жели да испланира свој обилазак и да уопште не губи време тамо. На интернету је истражила апсолутно све атракције, и свакој је доделила оцену колико јој се свиђа. Ипак, пошто атракција има много, а Ана мора и да спакује ствари, моли вас да јој помогнете у планирању обиласка.

Ана вам даје нову мапу Париза где су уписани сви пешачки путеви, од које до које атракције воде, и колико времена траје пешачење тим путем. Такође ће вам дати своје оцене атракција, и укупно време које ће провести у Паризу. Ваш задатак је да јој кажете **колика је највећа укупна лепота атракција које она може да обиђе** (укупна лепота је сума оцена сваке атракције коју обиђе), тако да се Ана креће само новим пешачким путевима. Ана свој обилазак може почети од било које атракције.

Имајте у виду да Ана када обилази атракције не губи ни мало времена разгледајући, њој је довољно само да се слика за Инстаграм и да иде даље. Управо зато, могуће је да неке атракције обиђе и по више пута, а да су јој и даље исто лепе као што су јој биле и на почетку (односно, уколико више пута обиђе исту атракцију, сваки пут ће се њена оцена додати на укупну лепоту пута).

## Опис улаза

У првој линији се налазе два природна броја  $N$  и  $T$  - број атракција, и време које ће Ана провести у Паризу. У другој линији се налази  $N$  природних бројева - оцене које је Ана дала атракцијама. У трећој линији се налази  $N$  природних бројева - низ  $X_i$ . У четвртој линији се налази  $N$  природних бројева - низ  $D_i$ . Низови  $X$  и  $D$  описују једносмерне пешачке путеве - пут који води до атракције  $i$  почиње од атракције  $X_i$ , и потребно је  $D_i$  времена да се препешачи. Атракције су индексирани бројевима од 1 до  $N$ .

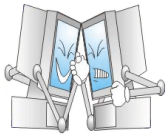
## Опис излаза

У први и једини ред излаза исписати један број - највећу укупну лепоту атракција које Ана може да обиђе.

## Пример 1

### Улаз

```
5 7
7 3 1 4 8
4 3 5 2 1
3 2 4 1 7
```



## Излаз

16

## Објашњење примера

У овом примеру постоји 5 атракција, а пешачки путеви који постоје су  $4 \rightarrow 1$  дужине 3,  $3 \rightarrow 2$  дужине 2,  $5 \rightarrow 3$  дужине 4,  $2 \rightarrow 4$  дужине 1 и  $1 \rightarrow 5$  дужине 7. Време којие ће Ана провести у Паризу је 7, и за то време највећа укупна лепота атракција које може да обиђе је 16 ако иде путем:  $5 \rightarrow 3 \rightarrow 2 \rightarrow 4$ .

## Пример 2

### Улаз

```
6 200
1 1 1 1 1 100
3 1 2 5 4 5
1 1 1 10 10 1
```

## Излаз

201

## Објашњење примера

Да би овде постигла највећу лепоту, Ана ће се вртети у круг између пар атракција. Једна могућност је да крене од 2, а пут би био  $2 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 1$

## Ограничења

- $2 \leq N \leq 10^5$
- $1 \leq T \leq 10^{12}$
- $1 \leq A_i \leq 10^6$
- $1 \leq D_i \leq 10^6$

Постоје четири подзадатака:

- Подзадатак 1 [22 поена]:  $N, T \leq 1000$ .
- Подзадатак 2 [17 поена]:  $N \leq 1000$ , и  $T \leq 10^{12}$ .
- Подзадатак 3 [14 поена]:  $D_i = 1$  за свако  $i$ , односно време потребно да се сваки пут препешачи је 1.
- Подзадатак 4 [23 поена]: Низ  $X$  је пермутација бројева од 1 до  $N$ , односно из сваке атракције креће тачно један пут.
- Подзадатак 5 [24 поена]: Нема додатних ограничења.