



CONCURSUL JUDEȚEAN "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Informatică

Clasa a V-a

Problema 1 – Nicovala lui Gruia

Pe o uliță îngustă din satul Bărăști, la capătul din deal, se afla cea mai veche fierărie din ținut. Acolo trăia și muncea Gruia Potcoavă, un fierar lat în umeri și scurt la vorbă, ale cărui mâini bătătorite știau să împlânzească fierul roșu ca jarul.

Gruia nu lucra după ochi, ca alți fierari. El credea că o unealtă bună se naște dintr-un număr anume de lovituri de ciocan — nici una mai mult, nici una mai puțin. De aceea, înainte să înceapă orice lucrare, Gruia scria pe o bucată de piele numărul de lovituri pe care urma să le dea.

Tu ești ucenic în fierăria lui Gruia. În fiecare zi primești câte o bucată de piele cu un număr n notat pe ea și trebuie să răspunzi la întrebările meșterului.

Cerința 1 ($c = 1$)

Gruia crede că o lovitură este perfectă dacă cifra care o reprezintă este un pătrat perfect nenul. Cifrele pătrate perfecte nenule sunt: 1, 4, 9. Afișează câte cifre ale lui n sunt pătrate perfecte nenule.

Cerința 2 ($c = 2$)

Gruia știe că o unealtă iese bine doar dacă munca este cumpănită — adică loviturile sunt distribuite egal între prima și a doua jumătate a lucrării.

Spunem că n este cumpănit dacă suma cifrelor din prima jumătate este egală cu suma cifrelor din a doua jumătate. Dacă n are număr impar de cifre, cifra din mijloc se ignoră. Afișează **DA** dacă n este cumpănit, sau **NU** în caz contrar.

Date de intrare

Fișierul de intrare *gruia.in* conține:

- Pe prima linie se citește numărul c ($c = 1$ sau $c = 2$), reprezentând cerința care trebuie rezolvată.
- Pe a doua linie se citește numărul natural n ($1 \leq n \leq 999.999.999$), reprezentând numărul de lovituri notat de Gruia.

Date de ieșire

Fișierul de ieșire *gruia.out* va conține:

- **dacă $c=1$** , numărul de cifre pătrate perfecte ale lui n .
- **dacă $c=2$** , textul **DA** sau **NU**.

Restricții și precizări:

- Pentru cerința 1 sunt 40% dintre teste, restul de 60% sunt pentru cerința 2.
- Timp de execuție 1s.

Exemple

gruia.in	gruia.out	Explicație
1 914823649	5	Cerința 1 Cifre pătrate perfecte: 9,1,4,4,9
2 3241	DA	Cerința 2 <i>Prima jumătate: $3+2 = 5$. A doua jumătate: $4+1 = 5$. Sumele sunt egale</i>
2 13342	Nu	Cerința 2 <i>Cifra din mijloc (3) se ignoră. Prima jumătate: $1+3 = 4$. A doua jumătate: $4+2 = 6$. Sumele sunt diferite</i>



CONCURSUL JUDEȚEAN "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Informatică

Clasa a V-a

Problema 2 – Fabrica de vise

În inima orașului Dulcilor, ascunsă după dealuri de zahăr și păduri de scorțișoară, se înalță Fabrica de Vise — o clădire veche de 10 secole, ale cărei ziduri sunt îmbibate cu mirosul bomboanelor din alte vremuri. Nimeni nu știe exact cine a construit-o, dar legenda spune că primul cofetar a semnat contractul cu visele înseși.

Azi, fabrica este condusă de bătrânul Trandafir Melișcă, moștenitorul unei lungi dinastii de cofetari. Cu părul alb ca zahărul pudră și ochelarii mereu aburiți de la ceaun, Trandafir cunoaște fiecare secret al fabricii — inclusiv pe cel mai important: magia se ascunde în numărul 10, la fel de vechi ca fabrica însăși.

În fiecare dimineață, pe poarta ruginită a fabricii intră n copii unul după altul, fiecare primind de la Trandafir un număr de bomboane magice. Tu ești noul ucenic al lui Trandafir și ai primit azi două misiuni. Citește cu atenție jurnalul și ajută-l!

Misiunea 1 ($c = 1$)

Trandafir crede că numărul 10 este numărul magic al fabricii, pentru că de atâtea secole există. El consideră că un copil este norocos dacă suma cifrelor numărului de bomboane primit este divizibilă cu 10.

Misiunea 2 ($c = 2$)

Trandafir știe că magia numărul 7 se transmite din copil în copil atunci când mai mulți copii intrați consecutiv pe poartă au primit fiecare un număr de bomboane divizibil cu 7. Cu cât lanțul magic e mai lung, cu atât vraja e mai puternică.

Date de intrare

Fișierul de intrare *fabrica.in* conține:

- Pe prima linie numărul c ($c = 1$ sau $c = 2$), reprezentând misiunea care trebuie îndeplinită.
- Pe a doua linie numărul natural n ($1 \leq n \leq 1000$), reprezentând numărul de copii intrați azi în fabrică.
- Pe a treia linie n numere naturale b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10000$), reprezentând numărul de bomboane primit de fiecare copil, în ordinea în care au intrat.

Date de ieșire

Fișierul de ieșire *fabrica.out* va conține:

- dacă $c=1$, numărul de copii norocoși.
- dacă $c=2$, pe prima linie lungimea celui mai lung lanț de copii consecutivi care au primit fiecare un număr de bomboane divizibil cu 7, iar pe a doua linie numărul de ordine al copilului cu care începe acest lanț. Dacă niciunul dintre copii nu a primit un număr divizibil cu 7, se afișează 0.

Restricții și precizări:

- $1 \leq n \leq 1000$
- $1 \leq b_i \leq 10000$
- Timp de execuție: 1 secundă

Exemple

Fabrica.in	fabrica.out	Explicație
1 7 37 10 55 460 6 1009 19	5	Cerința 1 <i>37 (3+7=10 ✓), 55 (5+5=10 ✓), 460 (4+6+0=10 ✓), 1009 (1+0+0+9=10 ✓), 19 (1+9=10 ✓) sunt copii norocoși</i>
2 7 7 14 3 7 21 7 5	3 4	Cerința 2 <i>cel mai lung lanț este format de copiii 4, 5, 6 (cu 7, 21 și 7 bomboane). Lanțul are lungime 3 și începe de la copilul numărul 4</i>



CONCURSUL JUDEȚEAN "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Informatică

Clasa a VI-a

Problema 1 –Turnurile Numerice

În cetatea **Numeria** există un șir de turnuri magice, numerotate de la 1 la N, fiecare având înscris pe el un număr natural.

Vrăjitorii cetății studiază anumite proprietăți speciale ale turnurilor și au stabilit două tipuri de misiuni.

Două turnuri vecine se consideră **compatibile** dacă diferența absolută dintre valorile lor este un număr par.

Exemplu:

- 14 și 20 sunt compatibile deoarece $|14-20| = 6$, iar 6 este par.
- 15 și 22 nu sunt compatibile deoarece $|15-22| = 7$, iar 7 este impar.

Cerința 1

Să se determine lungimea celei mai mari secvențe consecutive de turnuri compatibile două câte două. Mai exact, într-o secvență validă: fiecare două turnuri vecine din secvență trebuie să fie compatibile.

Cerința 2

Pentru fiecare secvență consecutivă validă formată din cel puțin 2 turnuri: se calculează suma valorilor turnurilor din secvență. Să se determine suma maximă care poate fi obținută. Dacă nu există nicio secvență validă de lungime cel puțin 2, se va afișa 0.

Date de intrare

Fișierul **turnuri.in** conține:

- pe prima linie două numere naturale C și N, unde:
 - C reprezintă cerința (1 sau 2);
 - N reprezintă numărul de turnuri;
- pe a doua linie se află N numere naturale reprezentând valorile turnurilor.

Date de ieșire

Fișierul **turnuri.out** va conține:

- **dacă C = 1** Se va afișa lungimea maximă a unei secvențe valide.
- **dacă C = 2** Se va afișa suma maximă obținută pentru o secvență validă.

Restricții și precizări

- $1 \leq N \leq 100000$
- valorile din șir sunt numere naturale mai mici decât 10000
- o secvență formată dintr-un singur element NU este considerată validă pentru cerința 2
- pentru 40% dintre teste: $N \leq 1000$
- timpul recomandat de execuție: 1 secundă

Exemplul 1	Exemplul 2
turnuri.in 1 8 4 10 16 11 7 3 8 12 turnuri.out 3	turnuri.in 2 8 4 10 16 11 7 3 8 12 turnuri.out 30
Explicație Secvențele compatibile sunt: <ul style="list-style-type: none"> • (4,10,16) – diferențele sunt 6 și 6 • (11,7,3) – diferențele sunt 4 și 4 • (8,12) – diferența este 4 Lungimea maximă este 3.	Explicație Secvența (11,7,3) este validă și are suma 21. Secvența (4,10,16) este validă și are suma 30. Secvența (8,12) are suma 20. Valoarea maximă este 30.



CONCURSUL JUDEȚEAN "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Informatică

Clasa a VI-a

Problema 2 – Antrenamentul lui Mihai

Pe stadionul din Târgu-Frumos, în fiecare dimineață înainte de răsărit, **Mihai Dobrescu** își face antrenamentul de alergare. Mihai nu aleargă ca toată lumea — el nu caută să fie constant, ci să fie viu. Aleargă o tură repede, una mai lentă, una iar repede — ca și cum inima lui bate în ritm de munte.

Antrenorul său, bătrânul **Fănică**, cronometrează fiecare tură și notează timpii într-un carnet. La sfârșitul antrenamentului, Fănică analizează carnetul și vrea să afle două lucruri despre performanța lui Mihai.

Tu ești asistentul lui Fănică. Ia carnetul și ajută-l!

Cerința 1

Fănică vrea să știe cât de mult variază efortul lui Mihai de la o tură la alta. El calculează **amplitudinea antrenamentului** — diferența dintre cel mai lent timp și cel mai rapid timp din întregul antrenament. **Afișează** diferența dintre timpul maxim și timpul minim din șir.

Cerința 2 (c = 2)

Fănică știe că Mihai aleargă cel mai bine în **ritm zig-zag** — alternând ture rapide cu ture lente. O secvență de ture este **zig-zag** dacă timpii alternează strict: fiecare tură este fie strict mai mare, fie strict mai mică decât cea anterioară, și sensul se schimbă la fiecare pas.

Mai formal, o secvență t_i, t_{i+1}, \dots, t_j este zig-zag dacă pentru orice k cu $i \leq k < j$:

— dacă $t_k < t_{k+1}$ atunci $t_{k+1} > t_{k+2}$

— dacă $t_k > t_{k+1}$ atunci $t_{k+1} < t_{k+2}$

Cu alte cuvinte, fiecare element interior este fie **strict mai mare** decât ambii vecini, fie **strict mai mic** decât ambii vecini. Nu sunt permise două diferențe consecutive de același sens, și nici elemente egale. **Afișează** pe prima linie lungimea celei mai lungi secvențe zig-zag, iar pe a doua linie indicele din șir al primului ei element (indexat de la 1). Dacă există mai multe secvențe de lungime maximă, se afișează cea care începe mai devreme.

Date de intrare

Fișierul de intrare *stadion.in* conține

- Pe prima linie, numărul c ($c = 1$ sau $c = 2$), reprezentând cerința care trebuie rezolvată.
- Pe a doua linie, numărul natural n ($2 \leq n \leq 100.000$), reprezentând numărul de ture alergiate.

- Pe a treia linie, n numere naturale t_1, t_2, \dots, t_n ($1 \leq t_i \leq 3600$), reprezentând timpul în secunde al fiecărei ture.

Date de ieșire

Fișierul de ieșire *stadion.out* va conține:

- Dacă $c=1$, diferența dintre timpul maxim și timpul minim din șir
- Dacă $c=2$, pe prima linie lungimea celei mai lungi secvențe zig-zag, iar pe a doua linie indicele din șir al primului ei element (indexat de la 1). Dacă există mai multe secvențe de lungime maximă, se afișează cea care începe mai devreme.

Restricții și precizări:

- $1 \leq n \leq 100.000$
- $1 \leq t_i \leq 3600$ secunde
- Timp de execuție: 1 secundă
- Pentru 40% din teste: $c = 1$ | Pentru 60% din teste: $c = 2$

Exemplul 1	Exemplul 2
stadion.in 1 6 42 55 38 50 40 56 stadion.out 18	stadion.in 2 8 40 45 38 50 42 48 48 51 stadion.out 6 1
Explicație Explicație: minim = 38, maxim = 56, Diferența: $56 - 38 = 18$	Explicație secvența 40 45 38 50 42 48 (pozițiile 1-6) este zig-zag: 40<45>38<50>42<48, lungime 6. La poziția 7 avem 48 48 — egale, deci secvența se rupe. Cea mai lungă secvență are lungime 6 și începe la poziția 1.



CONCURSUL JUDEȚEAN "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Informatică

Clasa a VII-a / a VIII-a

Problema 1 – Rețea energetică

În anul 2140, orașul **NeoGrid** este alimentat de o rețea inteligentă formată din celule energetice dispuse sub forma unei matrice pătratice de dimensiune $n \times n$. Fiecare celulă conține un nivel energetic reprezentat printr-o cifră între 0 și 9. Două celule aparțin aceleiași **zone energetice** dacă:

- au aceeași valoare;
- se pot conecta prin deplasări succesive pe direcțiile sus, jos, stânga sau dreapta, trecând doar prin celule cu aceeași valoare.

Centrul de control al orașului dorește să analizeze stabilitatea energetică a rețelei.

Cerința 1

Să se determine **numărul total de celule** din matrice care au **valoarea maximă** (cea mai mare cifră prezentă în matrice).

Cerința 2

Pentru fiecare cifră d din intervalul $[0, 9]$, se definește: $P(d)$ = suma dimensiunilor tuturor zonelor energetice formate din valoarea d .

Să se determine:

- cifra d pentru care valoarea $P(d)$ este maximă;
- valoarea maximă obținută.

Dacă există mai multe cifre cu aceeași valoare maximă, se va afișa cifra cea mai mică.

Date de intrare

Fișierul **retea.in** conține:

- pe prima linie, numărul natural c (1 sau 2) reprezentând cerința;
- pe a doua linie, numărul natural n ;
- pe următoarele n linii, câte n numere naturale din intervalul $[0, 9]$, reprezentând matricea energetică.

Date de ieșire

Fișierul **retea.out** va conține:

- **dacă $c = 1$** , Se va afișa pe prima linie numărul total de celule din matrice care au valoare maximă.

- **dacă $c = 2$** , Se vor afișa două numere: cifra pentru care $P(d)$ este maxim și valoarea maximă cu un spațiu între ele.
-

Restricții și precizări

- $1 \leq n \leq 500$
- două celule sunt vecine doar dacă au latură comună;
- pentru 40% dintre teste: $n \leq 100$;
- timpul de execuție recomandat: 1 secundă.

Exemplul 1	Exemplul 2
retea.in 1 5 1 1 2 3 3 1 2 2 3 4 5 5 2 4 4 5 6 6 4 7 8 8 8 7 7 retea.out 3	retea.in 2 5 1 1 2 3 3 1 2 2 3 4 5 5 2 4 4 5 6 6 4 7 8 8 6 7 7 retea.out 2 4
Explicație: Cea mai mare valoare din matrice este 8 și apare de 3 ori.	Explicație: Valorile 2 și 4 au aceeași dimensiune energetică 4. Se va alege cea mai mica valoare adică 2.



CONCURSUL JUDEȚEAN "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Informatică

Clasa a VII-a / a VIII-a

Problema 2 – Labirintul Portalurilor

Într-un laborator secret există un labirint reprezentat printr-o matrice cu n linii și m coloane.

Fiecare poziție poate avea una dintre valorile:

- 0 – spațiu liber;
- 1 – zid;
- 2 – portal.

Un robot experimental pornește din poziția (1, 1) și trebuie să ajungă în poziția (n, m). Robotul se poate deplasa doar:

- sus;
- jos;
- stânga;
- dreapta,

fără a traversa ziduri sau a ieși din matrice.

Funcționarea portalurilor

Toate portalurile sunt numerotate în ordinea apariției lor în matrice (linie cu linie, de la stânga la dreapta). Dacă robotul intră într-o celulă portal:

- este teleportat instantaneu la următorul portal din ordine;
- dacă portalul curent este ultimul, robotul rămâne pe poziție.

Teleportarea nu consumă mutări suplimentare.

Cerința 1

Să se determine **numărul de celule libere (valoarea 0)** din matrice.

Cerința 2

Să se determine numărul drumurilor distincte de lungime minimă prin care robotul poate ajunge la destinație. Două drumuri sunt distincte dacă diferă prin cel puțin o poziție vizitată. Dacă nu există drum, se va afișa -1.

Date de intrare

Fișierul *portal.in* conține:

- pe prima linie, valoarea c (1 sau 2);
- pe a doua linie, numerele naturale n și m ;
- pe următoarele n linii, câte m valori din mulțimea $\{0,1,2\}$.

Date de ieșire

Fișierul *portal.out* va conține:

Dacă $c = 1$ Numărul de celule libere (valoare 0) din matrice.

Dacă $c = 2$ Numărul drumurilor minime distincte.

Restricții și precizări

- $1 \leq n, m \leq 300$
- există cel mult 1000 de portaluri;
- pozițiile (1, 1) și (n, m) sunt diferite de 1;
- pentru 30% dintre teste nu există portaluri;
- timpul maxim recomandat: 1 secundă.
-

Exemplul 1	Exemplul 2
portal.in 1 4 5 0 0 1 0 1 1 0 1 0 2 0 0 2 0 0 1 0 0 0 portal.out 13	portal.in 2 3 4 0 0 2 0 1 0 1 0 2 0 0 0 portal.out 2
Explicație Numărăm celulele cu valoarea 0: <ul style="list-style-type: none"> • Linia 1: 4 zerouri • Linia 2: 2 zerouri • Linia 3: 3 zerouri • Linia 4: 4 zerouri Total = 13.	Explicație Există două drumuri distincte de lungime minimă prin care robotul poate ajunge la destinație.



CONCURSUL "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Disciplina Informatică

Clasa a IX-a

Problema 1 Bipătrate

Limită de timp	Memorie
2 secunde	1024 MiB

Enunț

Un număr întreg pozitiv X este numit **număr bipatrat** dacă și numai dacă există o pereche de numere întregi pozitive

(a, b) astfel încât:

$$X = 2^a \times b^2$$

De exemplu, **400** este un număr bipătrat deoarece $400 = 2^2 \times 10^2$.

Dat fiind un număr întreg pozitiv N , rezolvați cele două cerințe de mai jos.

Cerința 1: Găsiți cel mai mare număr bipătrat din intervalul $[1, N]$.

Cerința 2: Determinați câte numere bipătrate există în intervalul $[1, N]$.

Date de intrare

Pe prima linie a fișierului de intrare se citește numărul întreg C (numărul cerinței) și apoi numărul întreg N , separate printr-un spațiu.

Date de ieșire

Dacă $C = 1$, pe prima linie a fișierului de ieșire se va scrie cel mai mare număr bipătrat din intervalul $[1, N]$.

Dacă $C = 2$, pe prima linie a fișierului de ieșire se va scrie câte numere bipătrate există în intervalul $[1, N]$.

Restricții

- $1 \leq N \leq 10^{18}$
- N este număr întreg
- $1 \leq C \leq 2$
- **Atenție:** N poate să nu încapă într-un întreg pe 32 de biți!

Punctaj

Cerința 1 valorează 40 de puncte, iar Cerința 2 valorează 60 de puncte.

Exemple

Cerința 1 (C = 1)

Exemplul 1

bipatrate.in	bipatrate.out
1 20	18

Explicație: Cel mai mare număr bipătrat din intervalul $[1, 20]$ este **18** ($18 = 2^1 \times 3^2$).

Exemplul 2

bipatrate.in	bipatrate.out
1 399	392

Explicație: Cel mai mare număr bipătrat din intervalul $[1, 400]$ este **392** ($392 = 2^3 \times 7^2$).

Cerința 2 (C = 2)

Exemplul 1

bipatrate.in	bipatrate.out
2 20	5

Explicație: Numerele bipătrate din intervalul $[1, 20]$ sunt: **2, 4, 8, 16, 18** (în total 5).

Exemplul 2

bipatrate.in	bipatrate.out
2 400	24

Exemplul 3

bipatrate.in	bipatrate.out
2 1234567890	42413



CONCURSUL "VIOREL SADOVEANU"
Ediția a V-a - 23.05.2026
Disciplina Informatică
Clasa a IX-a

Problema 2 - Cetate

Limita de timp: 2 secunde

Limita de memorie: 1024 MiB

Enunț

În Regatul CEX există N ziduri de cetate, numerotate de la 1 la N . De asemenea, există M tunuri.

Tunul i apără zidurile cu numerele de la L_i la R_i (inclusiv).

Când un tun este distrus, zidurile pe care le apără nu mai sunt protejate de acel tun.

Care este numărul minim de tunuri ce trebuie distruse astfel încât să existe cel puțin un zid de cetate care să nu fie apărat de niciun tun?

Restricții

- $1 \leq N \leq 10^6$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq L_i \leq R_i \leq N$, pentru orice $1 \leq i \leq M$
- Toate valorile din intrare sunt numere întregi.

Format de intrare

Datele de intrare sunt furnizate din fișierul *cetate.in* în următorul format:

```
N M  
  
L1 R1  
  
L2 R2  
  
⋮  
  
Lm Rm
```

Format de ieșire

Afișați în fișierul *cetate.out* numărul minim de tunuri ce trebuie distruse astfel încât să existe cel puțin un zid de cetate neapărat de niciun tun.

Exemple

Exemplul 1

<i>cetate.in</i>	<i>cetate.out</i>
10 4 1 6 4 5 5 10 7 10	1

Dacă tunul 1 este distrus, niciun tun nu mai apără zidul 3.

De asemenea, dacă niciun tun nu este distrus, toate zidurile sunt apărate de cel puțin un tun.

Prin urmare, răspunsul este 1.

Exemplul 2

<i>cetate.in</i>	<i>cetate.out</i>
5 2 1 2 3 4	0

Deoarece niciun tun nu apără zidul 5, există deja un zid neapărat fără a fi nevoie să distrugem vreun tun.

Prin urmare, răspunsul este 0.

Exemplul 3

<i>cetate.in</i>	<i>cetate.out</i>
5 10 2 5 1 5 1 2 2 4 2 2 5 5 2 4 1 2 2 2 2 3	3



CONCURSUL ”VIOREL SADOVEANU”

Ediția a V-a - 23.05.2026

Disciplina Informatică

Clasa a X-a

Problema 1 Tilda

Limita de timp	Memorie
2 secunde	1024 MiB

Enunț

Pentru un șir de numere întregi $A = (A_1, A_2, \dots, A/|A|)$, spunem că A este în **formă de tildă** dacă satisface simultan toate cele patru condiții de mai jos:

- Lungimea $|A|$ este cel puțin 4.
- $A_1 < A_2$.
- Există exact un indice întreg i cu $2 \leq i < |A|$ astfel încât $A[i-1] < A[i] > A[i+1]$ (maxim local).
- Există exact un indice întreg i cu $2 \leq i < |A|$ astfel încât $A[i-1] > A[i] < A[i+1]$ (minim local).

Cu alte cuvinte, șirul crește la început, atinge un singur vârf, apoi coboară până la o singură vale, după care crește din nou — exact ca forma literei \sim (tildă).

Se dă o permutare $P = (P_1, P_2, \dots, P_N)$ a mulțimii $\{1, 2, \dots, N\}$. Determinați numărul de subsiruri contigue ale lui P care sunt în formă de tildă.

Date de intrare

Prima linie a fișierului de intrare conține numărul întreg N .

A doua linie conține N numere întregi separate prin spații: P_1, P_2, \dots, P_N — elementele permutării.

Date de ieșire

Fișierul de ieșire va conține un singur număr întreg — numărul de subsiruri contigue ale lui P care sunt în formă de tildă.

Restricții și precizări

- $4 \leq N \leq 300\,000$
- P este o permutare a mulțimii $\{1, 2, \dots, N\}$ (toate elementele sunt distincte și aparțin intervalului $[1, N]$).
- Toate valorile din input sunt numere întregi.
- Se garantează că există cel puțin un subșir în formă de tildă.

Subtask	Punctaj	Condiții suplimentare
1	20 puncte	$N \leq 1\,000$ (soluție $O(N^2)$ acceptată)
2	30 puncte	$N \leq 50\,000$
3	50 puncte	$N \leq 300\,000$ (soluție eficientă $O(N)$ necesară)

Exemple

Exemplul 1

<i>tilda.in</i>	<i>tilda.out</i>
6 1 3 6 4 2 5	2

Explicație: Subsirurile contigue în formă de tildă sunt $(3, 6, 4, 2, 5)$ și $(1, 3, 6, 4, 2, 5)$.

Exemplul 2

<i>tilda.in</i>	<i>tilda.out</i>
6 1 2 3 4 5 6	0

Explicație: Șirul este strict crescător, nu există niciun subsir cu exact un maxim și un minim local, deci răspunsul este 0.

Exemplul 3

<i>tilda.in</i>	<i>tilda.out</i>
12 11 3 8 9 5 2 10 4 1 6 12 7	4



CONCURSUL "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Disciplina Informatică

Clasa a X-a

Problema 2 Continuu

Limita de timp	Memorie
2 secunde	1024 MiB

Enunț

Se dă un șir S de lungime N , format doar din caracterele 0 și 1 .

Poți efectua următoarea operație de oricâte ori (inclusiv de zero ori):

- Alege un număr întreg i cu $1 \leq i \leq N$ și schimbă S_i : dacă era 0 devine 1 , dacă era 1 devine 0 .

Scopul tău este să faci ca toate aparițiile lui 1 din S să formeze **cel mult un interval continuu**. Găsește numărul minim de operații necesare pentru a atinge acest scop.

Mai precis, scopul este să obții un șir S pentru care există o pereche de numere întregi (l, r) cu proprietățile de mai jos. Găsește numărul minim de operații necesare.

- $1 \leq l \leq r \leq N + 1$
- Pentru orice întreg i cu $1 \leq i \leq N$: $S_i = 1$ este echivalent cu $l \leq i < r$.

Se poate demonstra că scopul poate fi atins întotdeauna într-un număr finit de operații.

Se dau T cazuri de test; rezolvă fiecare dintre ele.

Restricții

- $1 \leq T \leq 20.000$
- $1 \leq N \leq 2 \times 10^5$
- S este un șir de lungime N format doar din 0 și 1 .
- Suma valorilor N pentru toate cazurile de test dintr-un fișier de intrare este cel mult 2×10^5 .
- T și N sunt numere întregi.

Date de intrare

Datele de intrare sunt citite din fișierul *continuu.in* în formatul următor:

```
T
caz_1
caz_2
...
caz_T
```

Cazul i este reprezentat astfel:

```
N
S
```

Date de ieșire

Afișează în *continuu.out* T linii. Linia i ($1 \leq i \leq T$) trebuie să conțină răspunsul pentru cazul de test i .

Exemple

Exemplul 1

<i>continuu.in</i>	<i>continuu.out</i>
3	1 0 0
5 10011	
10 1111111111	
7 0000000	

Explicații (Exemplul 1)

Cazul 1: Dacă efectuăm operația de schimbare a lui S_1 din **1** în **0**, obținem șirul 00011, în care toți de **1** formează un interval contiguu. Șirul inițial nu satisface condiția, deci răspunsul este **1**.

Cazul 2: Șirul nu conține niciun **0**, deci nu este nevoie de nicio operație. Răspunsul este **0**.

Cazul 3: Șirul nu conține niciun **1**, deci nu este nevoie de nicio operație. Răspunsul este **0**.

Exemplul 2

<i>continuu.in</i>	<i>continuu.out</i>
5	0 2 3 0 2
2 01	
10 1000010011	
12 111100010011	
3 111	
8 00010101	



CONCURSUL "VIOREL SADOVEANU"
Ediția a V-a - 23.05.2026
Disciplina Informatică
Clasele XI-XII

Problema 1 Diferența

Limita de timp	Memorie
2 secunde	1024 MiB

Cerința

Se dă un șir de numere întregi $A = (A_1, A_2, \dots, A_N)$ de lungime N și un număr întreg nenegativ D . Se dorește ștergerea unui număr **minim** de elemente din A , astfel încât șirul rămas B să satisfacă condiția:

$$|B_i - B_j| \neq D, \text{ pentru orice } 1 \leq i < j \leq |B|$$

Determinați numărul **minim** de elemente care trebuie șterse.

Date de intrare

Pe prima linie se află două numere întregi, N și D .

Pe a doua linie se află N numere întregi, A_1, A_2, \dots, A_N , separate prin spații.

Date de ieșire

Pe prima linie se va afișa un singur număr întreg, reprezentând numărul minim de elemente care trebuie șterse din șirul A .

Restricții și precizări

Constrângere	Valori
$1 \leq N \leq 2 \times 10^5$	Numărul de elemente ale șirului
$0 \leq D \leq 10^6$	Diferența interzisă
$0 \leq A_i \leq 10^6$	Valorile elementelor șirului
Toate valorile	sunt numere întregi

Exemple

Exemplul 1

diferenta.in	diferenta.out
5 2 3 1 4 1 5	1

Explicație: Se șterge elementul $A_1 = 3$. Șirul rămas $B = (1, 4, 1, 5)$ satisface condiția: nu există două elemente cu diferența absolută egală cu 2.

Exemplul 2

diferenta.in	diferenta.out
4 3 1 6 1 8	0

Explicație: Șirul inițial satisface deja condiția. Nu este necesar să se șteargă niciun element.

Exemplul 3

diferenta.in	diferenta.out
10 3 1 6 2 10 2 3 2 10 6 4	2

Explicație: Se pot șterge, de exemplu, elementele cu valorile 3 și 1. Șirul rămas nu conține două elemente cu diferența egală cu 3.



CONCURSUL "VIOREL SADOVEANU"

Ediția a V-a - 23.05.2026

Disciplina Informatică

Clasele XI-XII

Problema 2 Graf 2-Regular

Limita de timp	Memorie
2 secunde	1024 MiB

Enunț

Se dă un graf neorientat simplu cu N noduri și M muchii.

Puteți efectua următoarele operații:

- adăugarea unei muchii;
- ștergerea unei muchii.

Fiecare operație costă 1.

Un graf se numește 2-regular dacă fiecare nod are exact gradul 2.

Cerință

Determinați numărul minim de operații necesare pentru a transforma graful dat într-un graf 2-regular.

Restricții

- $3 \leq N \leq 8$
- $0 \leq M \leq N(N-1)/2$
- Graful G dat in intrare este un graf simplu neorientat
- Toate valorile din intrare sunt numere întregi

Date de intrare

Datele de intrare sunt citite de la intrarea standard in urmatorul format:

```
N M
A1 B1
...
AM BM
```

Date de iesire

Afiseaza raspunsul.

Exemplul 1:

graf2.in	graf2.out
<pre>5 4 1 2 1 5 2 4 4 5</pre>	<pre>3</pre>

De exemplu, următoarele trei operații transforma G într-un graf simplu neorientat în care toate varfurile au gradul 2:

- Adauga muchia dintre varfurile 2 și 3.
- Sterge muchia dintre varfurile 2 și 4.
- Adauga muchia dintre varfurile 3 și 4.

graf2.in	graf2.out
3 0	3

graf2.in	graf2.out
6 8 1 4 1 5 2 3 2 6 3 4 3 6 4 5 4 6	2

graf2.in	graf2.out
8 21 1 4 5 7 8 4 3 4 2 5 8 1 5 1 2 8 2 1 2 4 3 1 6 7 5 8 2 7 6 8 5 4 3 8 7 3 7 8 5 3 7 4	13