Zadanie celoštátneho kola súťaže ZENIT v programovaní

Kategória A a B, 20-21.3.2024

V prípade nejasností konzultujte záložku **Pomoc** na stránke `zenit.ksp.sk`, alebo sa spýtajte organizátorov. Úlohy sú hodnotené úplne nezávisle a samostatne, takže ich môžete riešiť v ľubovoľnom poradí. Časový limit označuje, koľko času dostane váš program pri testovaní na našom serveri (nie na vašom lokálnom počítači). Počas súťaže môžete nájsť zadania aj na webstránke. Ak by sa papierové a tlačené zadania v nejakom detaile (napríklad časovom limite) nezhodovali, tak pravdu majú zadania na **webstránke**.

A: Aké to bolo číslo? — 10 bodov

Marek sa nedávno stavil so Zemanom, že dokáže zjesť veľa čokolády za jeden mesiac a pritom nepribrať. Jediný problém je, že zabudol, koľko presne čokolády to má zjesť. Našťastie si pamätá niekoľko množstiev, o ktoré sa určite nestavili.

Keďže Marek nechce prehrať stávku iba preto, že zabudol, o čo sa vlastne stavil a Zeman mu nepovie ani mäkké f, obrátil sa na vás. Vašou úlohou je nájsť nejaké množstvo čokolády, o ktoré sa mohol Marek staviť.

Vstup a výstup

Vstup pozostáva z dvoch riadkov. V prvom dostanete číslo n, $(1 \leq n \leq 1000)$, počet čísel, o ktoré sa Marek určite nestavil. Na druhom riadku dostanete medzerami oddelené tieto čísla. Všetky čísla na vstupe budú kladné a menšie ako 5000.

Na výstup vypíšte jedno celé kladné číslo menšie ako 5000[1], nejaké množstvo čokolády, o ktoré sa mohol Marek staviť. Ak takých čísel existuje viac, môžete vypísať ktorékoľvek. Ak žiadne také číslo neexistuje, vypíšte -1.

Príklad

vstup

```
10
1 4 7 2 5 6 6 1 5 3
```

výstup

```
47
```

B: Back to the... Future? — 15 bodov

Firma Zen IT vie úspešne prebrať iba také zásielky, ktoré obsahujú počet vecí rovný nejakej mocnine dvojky. Aby firma ochránila svojich zamestnancov od nekonečnej frustrácie, zakúpila si automatické zariadenie, ktoré vie overiť, či nie je zásielka problematická. Jediným problémom je, že sa včera pokazilo. Zhodou okolností predvčerom vypršala záruka.

Vrámci šetrenia sa predstavenstvo firmy rozhodlo namiesto certifikovaného opravcu povolať práve vás. Po krátkom vŕtaní sa v mašine ste zistili, že sa pokazila práve detekcia toho, či je číslo mocninou dvojky. Našťastie, ešte stále funguje tá časť, ktorá počet predmetov v zásielke prevedie do binárnej sústavy.

Pokiaľ sa vám podarí opraviť toto zariadenie, odmena vás isto neminie.

Vstup a výstup

Na jedinom riadku vstupu dostanete reťazec pozostávajúci z núl a jednotiek, počet predmetov v zásielke, zapísaný v binárnej sústave.

Ak je toto číslo mocninou dvojky, vypíšte dva riadky, v prvom z nich reťazec "`ANO`" (bez úvodzoviek), v druhom jedno číslo - koľká mocnina dvojky to je.

Ak toto číslo nie je mocninou dvojky, vypíšte tiež dva riadky, v prvom z nich reťazec "`NIE`" (tiež bez úvodzoviek), v druhom jedno číslo - koľkokrát môžeme toto číslo bezo zvyšku vydeliť dvomi.

Reťazec bude obsahovať nanajvýš 100 000 znakov. Môžete predpokladať, že reťazec na vstupe nebude mať hodnotu 0.

[1] Nulu nepovažujeme za kladné číslo, 5000 nie je menšie ako 5000 a teda nebude považované za správnu odpoveď.

Príklad

vstup	výstup
`101100`	`NIE` `2`

Vstupný reťazec kóduje číslo 44. Keď ho raz vydelíme dvomi, dostaneme 22. Keď ho vydelíme opäť dvomi, dostaneme číslo 11, ktoré už nie je deliteľné dvomi bezo zvyšku.

vstup	výstup
`0000000000001`	`ANO` `0`

C: Cukríkové dobrodružstvo

20 bodov

Keď šiel Bonifác minule zo školy, stala sa mu zvláštna vec; akýsi podivný pán mu ponúkol cukríky. Ale malo to jednu podmienku: Bonifác si mohol vybrať z niekoľkých kôp cukríkov tri, a to práve tak, aby bol súčet cukríkov v týchto kopách deliteľný troma. Bonifácovi ste naposledy pomohli a ten natešený bežal s cukríkmi za svojimi kamarátmi. Rozpovedal im túto historku. Hortenzia si však poťukala po čele.

"Bonifác, Bonifác, si ty ale chmuľo! Veď si určite mohol zobrať tých cukríkov viac, a ty si schmatol tri hocijaké kopy!"

"Keď si taká múdra, choď si tam sama," bránil sa Bonifác.

Hortenzia nie je žiaden mäkkýš. Hneď sa zdvihla, našla toho pána, ten jej ponúkol cukríky, všetko ako Bonifác popísal. Teraz však stojí pred ťažkou úlohou; ako má vybrať tri kopy cukríkov tak, aby súčet cukríkov v kopách bol čo najväčší, ale stále deliteľný troma? Nechce vyzerať pred kamarátmi zle. Pomôžete jej?

Vstup a výstup

Na vstupe dostanete v prvom riadku číslo n - počet kôp s cukríkmi. ($1 \leq n \leq 3000$) V druhom riadku bude n prirodzených čísel - počty cukríkov na jednotlivých kopách. Počet cukríkov na každej kope neprekročí 10^8.

Vyberte tri kopy cukríkov tak, aby súčet cukríkov bol deliteľný troma a zároveň bol čo najväčší. Na výstup vypíšte vybrané kopy cukríkov. Môžete predpokladať, že existuje aspoň jedno správne riešenie.

Príklad

vstup	výstup
`5` `1 15 4 7 2`	`15 7 2`

vstup	výstup
`4` `1 2 4 7`	`1 4 7`

D: Dosť bolo zdĺhavých rozprávok

20 bodov

V tejto úlohe sa budeme venovať postupnosti čísel, ktoré sa nazývajú Fibonacciho čísla. Fibonacciho čísla sú definované nasledovne:

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

Prvých niekoľko Fibonacciho čísel vyzerá nasledovne: $0, 1, 1, 2, 3, 5, 8, 13, 21...$

Vstup a Výstup

Na vstupe je jediné Fibonacciho číslo n, nepresahujúce 10^{18}.

Vypíšte ľubovoľné tri Fibonacciho čísla tak, aby ich súčet bol rovný n. Ak sa to nedá, vypíšte namiesto troch čísel vetu "`Nez odovzdam riesenie s touto odpovedou, poriadne si precitam a pochopim zadanie!`" bez úvodzoviek.

Uistite sa, že používate dostatočne veľkú celočíselnú premennú na ukladanie vstupu (`long long` v C++).

Príklad

vstup	výstup
121393	75025 17711 28657

E: Environmentálna kampaň KSP

25 bodov

Určite ste už počuli o globálnom otepľovaní. V tejto dobe je to všeobecne uznávaný fenomén, aj keď sa stále nedohodlo, nakoľko veľmi k nemu prispievajú ľudia.

Na rozdiel od vás, však o globálnom otepľovaní v Krajine Studeného Počasia ani nechyrovali. To sa zmenilo vtedy, keď sa dopočuli o tom, že kúrenie fosílnymi palivami bude v blízkej budúcnosti zakázané v snahe spomaliť klimatické zmeny.

To by samozrejme bola pre KSPákov katastrofa - v Krajine Studeného Počasia sa v zime bez kúrenia nezaobídu a do zelenej energie veru investovať nebudú. Ostáva už len jediné - zorganizovať environmentálnu kampaň a presvedčiť zvyšok sveta, že globálne otepľovanie vlastne neexistuje.

Ako na to? Jednoducho. Stačí len ukázať, že teploty z roka na rok nerastú. V KSP už osadníci N rokov každé leto merajú najvyššiu teplotu. Napísali si tieto teploty jednu za druhou a teraz ako dôkaz globálneho neotepľovania hľadajú také dva roky, že v tom nedávnejšiom z nich bola v lete nižšia maximálna teplota, ako v tom dávnejšiom.

Svoju environmentálnu kampaň by chcela KSP spustiť čo najskôr. Pomôžte im rýchlo takéto dva roky nájsť.

Vstup a Výstup

V prvom riadku je číslo N: počet meraní teploty. V druhom riadku je N čísel, predstavujúce najvyššiu letnú teplotu v KSP za posledných N rokov, od najdávnejšieho až po najnedávnejší. Platí $1 \leq N \leq 10^5$, $1 \leq teplota \leq 10^9$

Vypíšte dve čísla rokov i, j aby platilo $1 \leq i < j \leq N$ a zároveň teplota v roku i bola vyššia ako v roku j. Ak je viacero dvojíc spĺňajúcich tieto podmienky, vypíšte ľubovoľnú. Ak však takáto dvojica rokov neexistuje, vypíšte $-1 -1$.

Príklad

vstup	výstup
2 47 21	1 2

vstup	výstup
2 47 48	-1 -1

F: Fraktály

25 bodov

Fraktály sú matematické obrázky, v ktorých môžeme vidieť opakujúci sa vzor. V tejto úlohe si vyskúšate nakresliť zopár takýchto fraktálov zložených z dvoch znakov `'#'` (mriežka) a `' '` (medzera).

Náš fraktál bude charakterizovaný vzorom a číslom generácie. Vzor je štvorec zložený z medzier a mriežok. Vzor pre štvorec so stranou dĺžky tri môže vyzerať napríklad takto:

```
###
# #
###
```

Povieme si, že fraktál nultej generácie je iba znak `'#'` (mriežka). Nech dĺžka strany vzoru je s. Vo všeobecnosti platí, že fraktál $n+1$ generácie vznikne z n-tej generácie tak, že zoberieme $s \times s$ mriežku a do každého políčka mriežky položíme jednu kópiu fraktálu n-tej generácie, ak sa vo vzore na danom políčku nachádza mriežka. Ak sa vo vzore na danom mieste nachádza medzera, na dané miesto umiestnime rovnako veľký štvorec pozostávajúci iba z medzier.

Fraktál prvej generácie pre vzor spomenutý vyššie vyzerá takto[2]:

[2] Medzi každými dvoma znakmi (medzera alebo mriežka) z fraktálu budeme písať jednu medzeru, aby to vyzeralo krajšie.

```
# # #
#   #
# # #
```

Všimnite si, že v strede je medzera (nedali sme tam fraktál nultej generácie), lebo vzor obsahuje v strede medzeru. Fraktál druhej generácie vyzerá takto:

```
# # # # # # # # #
#   # #   # #   #
# # # # # # # # #
# # #       # # #
#   #       #   #
# # #       # # #
# # # # # # # # #
#   # #   # #   #
# # # # # # # # #
```

Znova si v strede môžeme všimnúť prázdny štvorček, do ktorého sme nedali fraktál prvej generácie, lebo vzor má v strede medzeru.

Vstup

V prvom riadku sa nachádzajú dve celé čísla s, g – dĺžka strany vzoru a číslo generácie. Pričom platí, že $2 \leq s \leq 20$ a $0 \leq g \leq 9$. V každom zo zvyšných s riadkov sa nachádza po s znakov, dokopy tvoriace vzor fraktálu popísaný medzerami a mriežkami. Nezabudnite, že každý riadok je ukončený znakom konca riadka.

Parametre na vstupe budú zvolené tak, aby veľkosť výstupu nepresiahla 600 000 znakov.

Výstup

Na výstupe sa nachádza obrázok fraktálu, ktorý sme si definovali vyššie. Každé dva znaky fraktálu (medzera alebo mriežka) oddeľte jednou medzerou navyše. Za posledným znakom v každom riadku už nedávajte medzeru.

Pozor, výstup v tejto úlohe je pomerne veľký a pomalé vypisovanie to nemusí stihnúť v čase. V Jave odporúčame použiť BufferedWriter, v Pythone vypisovať celé reťazce a použiť vhodné dátové štruktúry na reprezentáciu meniaceho sa reťazca (napr. bytearray).

Príklad

vstup

```
3 0
###
# #
###
```

výstup

```
#
```

vstup

```
3 1
###
# #
###
```

výstup

```
# # #
#   #
# # #
```

vstup

```
3 2
###
# #
###
```

výstup

```
# # # # # # # # #
#   # #   # #   #
# # # # # # # # #
# # #       # # #
#   #       #   #
# # #       # # #
# # # # # # # # #
#   # #   # #   #
# # # # # # # # #
```

G: Gregorova domáca úloha

30 bodov

Gregor mal dnes celkom zlý deň. Nielenže nepostúpil na krajské kolo Zenitu, ale ešte k tomu dnes na matematike zastupoval jeho najneobľúbenejší učiteľ. A veru aj dnes sa Gregorovi ušlo. Učiteľ napísal na tabuľu N čísel, a po dlhom dumavom pohľade na nich napokon s úškrnom zahlásil:

"Tak teda, deťúrence. Na domácu úlohu si dáme takéto neškodné cvičenie. Tu, hľa, som na tabuľu napísal veľa čísel a vy za úlohu musíte zistiť, ktoré čísla máme vynásobiť aby sme získali najväčší súčin. To by vás malo na dnes dobre zaneprázdniť."

Ach. Kto by už len dokázal vymyslieť takú nudnú, zdĺhavú a nepraktickú domácu úlohu. Keby len bol niekto, kto by Gregorovi pomohol a vyriešil ju namiesto neho.

Vstup a Výstup

V prvom riadku je číslo N: počet čísel. V druhom riadku je N čísel. Platí $1 \leq N \leq 10^5$, žiadne z nich v absolútnej hodnote neprekročí 10^9.

Vyberte niektoré spomedzi týchto čísel tak, aby ich súčin bol čo najväčší. Odpoveď budeme reprezentovať pomocou reťazca dĺžky N pozostávajúceho z jednotiek a núl, kde nula bude označovať, že toľké číslo do súčinu neberieme a jednotka bude označovať, že ho berieme. Nakoniec si ešte popíšeme exaktné pravidlá, ako má tento reťazec vyzerať:

- Aspoň jeden znak bude jednotka, `1`.
- Ak je na i-tej pozícii v reťazci `1`, použijeme i-te číslo v našom súčine. Tento súčin musí byť maximálny možný.
- Spomedzi všetkých takýchto reťazcov má váš najmenej jednotiek
- Spomedzi všetkých takýchto reťazcov je váš lexikograficky najväčší. Keď lexikograficky porovnávame dva reťazce, pozrieme sa na najľavšiu pozíciu, na ktorej sa líšia. Lexikograficky väčší reťazec bude mať na tejto pozícii väčší znak (jednotku).[3]

Navyše, v polovici vstupov je $N \leq 20$ a maximálny súčin sa zmestí do 64-bitovej premennej so znamienkom.

Príklad

vstup	výstup
`6` `-2 -2 -3 4 5 1`	`101110`

Najviac sa nám oplatí zobrať čísla $-2, -3, 4, 5$. Jednotku neberieme, aby sme neporušili tretí bod a spomedzi dvoch -2 si vyberieme tú ľavšiu, aby sme splnili aj štvrtý bod.

H: Hryzáky

30 bodov

Filoména našla v šuflíku veľmi starý školský model zubov. Po chvíli skúmania si všimla, že všetky zuby sú obyčajné rovnostranné trojuholníky s dĺžkou strany 1. Taktiež si všimla, že zuby sú podozrivo pravidelne uložené, konkrétne v dlhom rade pomyselných štvorčekov so stranou dĺžky 1, pričom niektoré trojuholníčky sa dotýkajú hornej strany pomyselného štvorčeka a niektoré dolnej. Všetko pohromade držalo pomocou jediného kusu veľmi pevného drôtu.

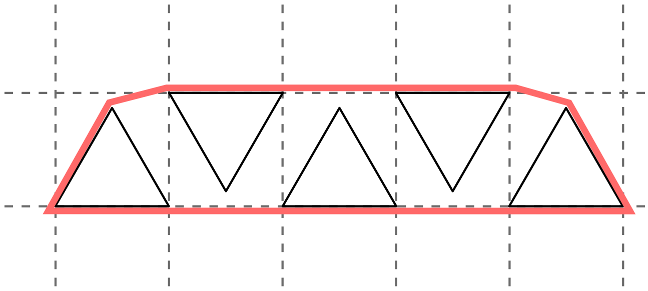
Filoména začala rozmýšľať, koľko najmenej pevného drôtu by potrebovala, aby podobný model vyrobila sama. Vedeli by ste jej to vypočítať?

Vstup a výstup

Na vstupe dostanete reťazec pozostávajúci zo znakov `D` a `H`, reprezentujúci zuby v modeli tak, ako nasledujú v rade. Znak `D` reprezentuje zub dotýkajúci sa dolnej strany svojho štvorca, znak `H` zub dotýkajúci sa hornej strany. Dĺžka reťazca nepresiahne 1000 znakov.

Vypíšte jediný riadok obsahujúci jedno reálne číslo - najmenšiu možnú dĺžku drôtu, ktorý by obkolesil všetky zuby. Za správne riešenie budeme považovať každé, ktorého absolútna alebo relatívna chyba neprekročí 10^{-6}. Dajte si pozor a vypíšte dostatočne veľa desatinných miest (aspoň 6, ideálne ešte viac).

[3] Voľne prerozprávané to znamená, že chcete mať jednotky čo najviac vľavo vo vašom reťazci.



Obr. 1: Príklad modelu zubov. Niektoré trčia dohora, niektoré dodola. Všetky zuby obkolesuje pevný drôt (znázornený slabou červenou farbou).

Príklad

vstup	výstup
`DHDHD`	`11.0352761804`

Tento vstup je znázornený hore na obrázku.

vstup	výstup
`DDD`	`7.00000000000`

vstup	výstup
`DDH`	`7.0236092694`

I: Ideme po poli

40 bodov

V ďalekej budúcnosti, v roku 2047, vynašli vedci nový spôsob dopravy. Ide o tzv. auto na diskrétny pohon. Cestovanie takýmto autom nestojí vôbec nič, ale má svoje obmedzenia. Napríklad takú rýchlosť nemôže zmeniť len tak kedykoľvek. S úderom každej sekundy ju môže zvýšiť o 1, znížiť o 1, alebo ponechať rovnakú.

Niektoré pravidlá cestnej premávky sa ešte nestihli prispôsobiť tomuto novému spôsobu dopravy. Stále sú oblasti, cez ktoré jednoducho nemôžete ísť príliš rýchlo. Zistiť, ako najrýchlejšie sa viete dostať z jedného miesta na druhé, môže byť pomerne náročné. Preto to teraz dostávate na úlohu vy!

Úloha

Cestu si rozdelíme na n bodov uložených na jednej priamke, susedné dva body majú vzdialenosť presne 1. V každom bode je určená maximálna rýchlosť. Treba ju dodržať buď, keď do bodu vchádzame, alebo keď z neho vychádzame (nie nutne oboje). Napríklad ak má bod maximálnu rýchlosť 5, môžeme doňho prísť rýchlosťou 5 a výjsť rýchlosťou 6, alebo prísť rýchlosťou 6 a výjsť rýchlosťou 5, ale nemôžeme príjsť aj výjsť rýchlosťou 6.

Špeciálny význam bude mať hodnota 0, ktorá znamená, že v danom bode musíme aspoň sekundu úplne stáť.

Vašou úlohou je zistiť, ako najrýchlejšie sa vieme dostať z prvého bodu do posledného bodu tak, že začíname s nulovou rýchlosťou a taktiež skončíme s nulovou rýchlosťou a máme dovolené hýbať sa iba dopredu.

Vstup a výstup

Na prvom riadku je číslo n, $(2 \leq n \leq 50\,000)$. Na druhom riadku je n čísel x_i, $(0 \leq x_i \leq 10^9)$, i-te z nich vyjadruje obmedzenie rýchlosti v i-tom bode.

Na výstup vypíšte jediné celé číslo, počet sekúnd, koľko trvá najrýchlejšia cesta z prvého bodu do posledného.

Pozor, časový limit je pomerne tesný, optimálne programy v pomalších jazykoch, ako aj rozšafné programy v rýchlejších, pravdepodobne zopár bodov stratia.

Príklad

vstup	výstup
`10` `1 2 3 4 5 6 7 8 9 10`	`5`

Rozpis po sekundách:

1. Rýchlosťou 1 prejdeme z prvého bodu do druhého.

2. Rýchlosťou 2 prejdeme z druhého bodu do štvrtého.
3. Rýchlosťou 3 prejdeme zo štvrtého bodu do siedmeho.
4. Rýchlosťou 2 prejdeme zo siedmeho bodu do deviateho.
5. Rýchlosťou 1 prejdeme z deviateho bodu do cieľového desiateho.

vstup

```
4
1 0 0 1
```

výstup

```
5
```

Po prvej sekunde sa dostaneme do druhého bodu, druhú sekundu v ňom stojíme. Tretiu sekundu sa dostaneme do tretieho bodu, štvrtú sekundu v ňom stojíme. Nakoniec v piatej sekunde sa dostaneme do posledného bodu, v ktorom na jej konci zastavíme.

vstup

```
10
0 1 2 2 3 3 2 2 1 0
```

výstup

```
5
```

Môžeme si dovoliť ísť presne rovnako, ako v prvom ukážkovom vstupe. Napríklad druhý bod má síce obmedzenie rýchlosti 1*, ale keďže doňho prichádzame rýchlosťou* 1*, odísť z neho môžeme kľudne aj rýchlosťou* 2*.*

vstup

```
10
0 1 2 3 2 2 3 2 1 0
```

výstup

```
6
```

V tomto prípade nemôžeme postupovať ako v prvom, lebo v rýchlosti 3 *by sme prešli až cez dva body (piaty a šiesty) s obmedzením rýchlosti na* 2*. Pôjdeme teda nasledovne (opäť rozpis po sekundách):*

1. Rýchlosťou 1 prejdeme z prvého bodu do druhého.
2. Rýchlosťou 2 prejdeme z druhého bodu do štvrtého.
3. Rýchlosťou 2 prejdeme zo štvrtého bodu do šiesteho.
4. Rýchlosťou 2 prejdeme zo šiesteho bodu do ôsmeho.
5. Rýchlosťou 1 prejdeme z ôsmeho bodu do deviateho.
6. Rýchlosťou 1 prejdeme z deviateho bodu do cieľového desiateho.

J: Javorová Polička

40 bodov

Všetci máme doma poličku. Čo sa stane s poličkou, ak ju zopár mesiacov necháme na pokoji? Presne tak, napadne na ňu vrstva prachu a potom vám mama nedá pokoja, kým sa ho nezbavíte. To zvyčajne nebýva problém - stačí cez poličku párkrát prejsť vlhkou handrou.

Veci však fungujú inak v Krajine Siahodlhých Poličiek. Tam majú totiž javorovú poličku dlhú až N metrov. Keďže táto starodávna polička je generácie uchovávaná vzácnosť KSP, aby sa neznesvätila, nikdy sa neutiera. Obyvatelia Krajiny Siahodlhých Poličiek si však uvedomili, že ak sa polička príliš zapráši, mohlo by ju to poškodiť. Namontovali teda pri každom metri poličky snímače, ktoré dokážu vyčísliť, koľko prachu na každý meter napadá.

Obyvatelia KSP budú tieto hodnoty monitorovať a budú potrebovať vedieť, aká najvyššia vrstva prachu sa nachádza na nejakom súvislom úseku poličky.

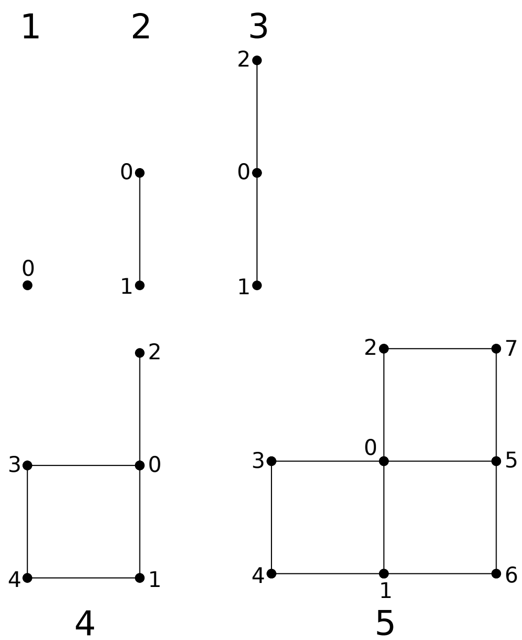
Vstup a Výstup

V prvom riadku sú čísla N, Q: dĺžka poličky a počet udalostí. V druhom riadku je N čísel: výška vrstvy prachu na každom z N metrov poličky.

Nasleduje Q riadkov v tvare o a b c. Ak $o = 0$, vypíšte výšku najvyššej vrstvy prachu na poličke od metra a po meter b. V tomto prípade je $c = 0$ a môžete ho ignorovať. Inak $o = 1$, a znamená, že na každý meter od a po b napadla vrstva prachu výšky c.

Vstup spĺňa nasledujúce obmedzenia:

- $1 \leq Q, N \leq 5 \cdot 10^5$



Obr. 2: Príklad prvých 5 Fibonacciho grafov.

- začiatočná vrstva prachu na každom metri bude celé číslo medzi 1 a 10^9
- $1 \leq a \leq b \leq N$.
- $c = 0$ ak $o = 0$, inak platí $1 \leq c \leq 10^9$

Príklad

vstup

```
5 3
1 2 3 4 5
0 2 4 0
1 2 3 2
0 2 4 0
```

výstup

```
4
5
```

K: Krotitelia grafov

50 bodov

V tomto kole ste sa už stretli s Fibonacciho číslami. Podobný postup vytvárania z posledných dvoch sa dá však aplikovať aj na iné objekty, než len čísla. V tejto úlohe sa pozrieme napríklad na (neoficiálne nazývané) Fibonacciho grafy.

Ako prvý graf označíme graf obsahujúci jediný vrchol s číslom 0. Ako druhý označíme graf s dvomi spojenými vrcholmi s číslami 0 a 1. Každý ďalší graf skonštruujeme z predošlých dvoch. Konkrétne N-tý nasledovne:

1. Vezmeme $(N-1)$-vý a $(N-2)$-hý graf a vrcholy s rovnakým číslom spojíme hranou
2. K číslam vrcholov pochádzajúcich z $(N-2)$-hého grafu pripočítame $(N-1)$-vé Fibonacciho číslo (rozmyslite si, že týmto zaručíme, že pre každé číslo od 0 po $F_N - 1$ budeme mať práve jeden vrchol s týmto číslom).

Pre potreby tejto úlohy určíme prvé dve Fibonacciho čísla nasledovne: $F_1 = 1$ a $F_2 = 2$.
Vašou úlohou bude zistiť, aká je najkratšia cesta medzi dvojicou vrcholov v niektorom Fibonacciho grafe.

Vstup a výstup

Vstup pozostáva z dvoch riadkov. Na prvom je číslo N, $(N \leq 1000)$ označujúce, koľký graf berieme. Na druhom riadku vstupu dostanete dve čísla a a b, $(0 \leq a, b < F_N)$.

Aspoň 30 bodov môžete získať za riešenie, ktoré stihne v časovom limite spracovať tie vstupy, v ktorých bude mať graf nanajvýš 10^6 vrcholov.

Pozor, čísla vyskytujúce sa vo vstupoch tejto úlohy budú veľké a nezmestia sa do bežných 32 a 64-bitových premenných.

Príklad

vstup

```
5
4 7
```

výstup

```
4
```

vstup

```
29
100000 200000
```

výstup

```
10
```

Toto je príklad najväčšieho grafu s menej ako 10^6 *vrcholmi.*