

Úlohy 2. kola letnej časti

Termín odoslania riešení tejto série je pondelok 18. mája 2015.

1. Zerg Bot 2

kat. Z; 0 b za popis, 15 b za program

Keď ste v prvej sérii pomohli pracovitým vedúcim KSP s donáškou kofoly, spohodlneli a stali sa z nich leniví vedúci KSP. Namiesto chodenia na prednášky sledovali pobehujúceho robota a zatiaľčo pili donesenú kofolu, na stoloch sa im kopili domáce úlohy z programovania.

A už ich je toľko, že ich vedúci nestíhajú spraviť sami. Je preto vašou morálnou povinnosťou im pomôcť. Je to predsa vaša chyba, že ste im pomohli prvýkrát a oni z toho zleniveli.

Našťastie, všetky domáce úlohy sa dajú transformovať na mapu pre robotika Zerga, takže stačí ak spravíte to čo minule. A po skončení série môžete dať vedúcim poriadnu prednášku o tom, ako byť produktívny a neprokrastinovať.

Úloha

Táto úloha je viac hra ako úloha, budete sa hrať s robotom a snažiť sa ho navigovať naprieč Matfyzom.

Dostanete prístup k programovaciemu rozhraniu robota, spolu s náhľadom na level. V tomto rozhraní môžete pomocou jednoduchého programovacieho jazyka napísať pre robota program. Po napísaní si program môžete hneď aj spustiť, a uvidíte ako sa robot vďaka nemu pohybuje.

Vašou úlohou bude napísať program, pomocou ktorého robot bez ujmy na obvodoch prejde bludiskom. Samozrejme mu musí vydržať batéria a stačiť pamäť.

Hra má niekoľko sérií, jednou z nich je aj Tutoriál. Ten je pre riešiteľov KSP-Z nebudovaný, ale odporúčame si ho prejsť, pomôže zžiť sa s jazykom :). Ak ste si nepreriešili úlohy minulých sérií, môžete si prejsť aj tie.

Celý návod k hre ako aj k programovaciemu jazyku nájdete v hlavnom menu hry, kliknutím na tlačítko Návod.

Odvzdávanie a bodovanie

Hra má dve súťažné série po 5 levelov. Za každý level v sérii *Prask 2.4* dostanete 1 bod a za každý level v sérii *Prask 2.5* dostanete 2 body. Levely môžete riešiť v ľubovoľnom poradí. Akonáhle spustíte správny program a robot sa s jeho pomocou dostane do cieľa, budú vám automaticky pripísané body za daný príklad. To znamená, že k tejto úlohe nie je nutné odovzdávať žiaden popis alebo tradičným spôsobom odovzdávať program.

Hru nájdete na stránke ksp.sk/specialne/ksp/32/3/1/.

Poznámka na záver

Keďže sa vám v minulej sérii tak darilo (a navyše sa ľahké príklady minuli), úlohy v tejto sérii sú mierne obtiažnejšie. Ak sa vám nedarí vyriešiť nejakú úlohu, skúste si (znovu) prečítať vzorové riešenia minulej série. Nájdete tam veľa trikov, ktoré sa dajú znovu použiť.

No a keďže sú úlohy ťažšie, je veľmi náročné ich zoradiť podľa stúpajúcej náročnosti, ako tomu bolo v minulej sérii. To čo my pokladáme za najťažšie, môže byť pre vás jednoduché a naopak. Pozrite si preto **všetky levely** a skúste sa nad nimi zamyslieť.

2. Zabezpečenie

kat. Z; 6 b za popis, 4 b za program

V centrále Kryptografickej Spravodajskej Pobočky berú bezpečnosť veľmi vážne. Nedávno zistili, že niektoré semináre (napríklad Korešpondenčný Seminár z Programovania) nezabezpečujú svoju komunikáciu dostatočne. Preto navrhujú použiť nový Kryptografický Šifrovací Protokol. Je založený na technológii, ktorú používali už v starovekom Ríme.

Text sa zašifruje tak, že každé písmeno textu sa nahradí písmenom, ktoré je v abecede o k pozícií ďalej. Napríklad, ak by sme mali $k = 3$, tak všetky A sa nahradia písmenom D , B sa nahradí E , C sa nahradí F . . . No a v prípade, že by sme potrebovali použiť písmeno, ktoré sa nachádza v abecede za písmenom Z , pokračujeme

opäť od začiatku abecedy. Písmeno X sa preto pri posunutí $k = 3$ nahradí za A , Y za B a Z za C . Medzery medzi slovami zostávajú nezmenené.

No a keďže toto šifrovanie bolo dobré pre Rímske impérium, musí byť dobré aj pre KSP. Avšak doba o pár (tisíc) rokov pokročila, preto skôr než KSP začne používať KŠP odporúčané KSP, musí byť jeho bezpečnosť otestovaná. Práve spúšťame najväčšie testovanie KŠP v histórii KSP¹. A na to potrebujeme práve teba.

Úloha

Tvojou úlohou je napísať program, ktorý bude dešifrovať text zašifrovaný pomocou KŠP.

Formát vstupu

V prvom riadku vstupu je celé číslo n , pričom platí $1 \leq n \leq 20\,000$. Nasleduje n riadkov, každý obsahuje text pozostávajúci z veľkých písmen anglickej abecedy a medzier, ktorý je určený na dešifrovanie. Dĺžka riadku nepresiahne 100 znakov. Je známe, že každý text, pred tým ako bol zašifrovaný, začínal *V MENE KSP*.

Formát výstupu

Pre každý riadok textu na vstupe vypíšte do samostatného riadku dešifrovaný text (taktiež tvorený veľkými písmenami anglickej abecedy a medzerami).

Príklad

vstup

```
3
W NFOF LTQ WTFUDJ NVTJB OPTJU SVAPWF TVLOF
V MENE KSP NEZABUDAJTE POUZIVAT SIFROVANIE
L CUDU AIF FHEWHQCELQDYU ZU IKFUH
```

výstup

```
V MENE KSP VSETCI MUSIA NOSIT RUZOVE SUKNE
V MENE KSP NEZABUDAJTE POUZIVAT SIFROVANIE
V MENE KSP PROGRAMOVANIE JE SUPER
```

V prvom riadku je posun $k = 1$, v druhom sa text vôbec nedešifroval, teda $k = 0$ a v treťom je $k = 16$.

Nápoveda

V prípade, že nevieš, ako spracovávať text, nasledujúce odstavce sú určené práve pre teba.

V programovacích jazykoch máme okrem číselných premenných typu `int`, `integer` aj premenné na ukladanie znakov, ktoré zvyknú byť označované `char`. Premenná typu `char` má v podstate uložené číslo od 0 po 256 v Pascale (od -128 po 127 v C++), pričom každému číslu zodpovedá nejaký znak: napríklad kde napríklad číslu 65 zodpovedá písmeno 'A', číslu 66 zodpovedá písmeno 'B' atď. – pozri si ASCII tabuľku na odkaze www.asciitable.com.

Nápoveda – PASCAL

Na načítanie jedného znaku zo vstupu do premennej typu `char` môžeme použiť funkciu `read(c)`. Pole znakov dĺžky 255 (`var s : array[0..255] of char`) sa nazýva `string` (na nulte pozícii uložená je dĺžka stringu, `s[1]` je prvý znak, `s[n]` je n -tý znak). Na načítanie celého riadku do premennej `s` typu `string` sa používa `readln(s)`. Dĺžku tohto stringu zistíme príkazom `length(s)`. Znak konca riadku sa nenačítajú. ASCII hodnotu daného znaku (jeho číselný kód) vieme zistiť pomocou `ord(c)`. Ak chceme získať znak z jeho kódu (opačný smer ako `ord`) použijeme `chr(cislo_znaku)`. Napríklad `writeln(ord('A'))` vypíše 65. Ak by sme napísali `writeln(chr(ord('A')))`, tak program vypíše A, pretože `chr` zoberie hodnotu znaku A vrátenú `ord` a vráti daný znak.

Nápoveda – C++, iostream

Premennú typu `char` môžeme považovať za jednobajtové číslo alebo za jeden znak. V prípade, že používame knižnicu `iostream` (`#include <iostream>`) môžeme jeden znak načítať nasledovne: `cin >> c`, kde `c` je typu `char`. Ak chceme znak vypísať: `cout << c`. Ak chceme vypísať jeho číslo: `cout << int(c)` – takto už program vie, že ten kus pamäte, čo považoval za `char`, má považovať za `int`. Na prácu s reťazcami znakov sa dá využiť typ `string`. Prístup k n -tému znaku stringu `s` je podobný ako v Pascale (**ale nie rovnaký**): `s[n-1]`. V C++ a podobných jazykoch sa totiž indexuje od nuly (takže prvý znak je `s[0]`).²

Keď chceme číslo vypísať ako znak, môžeme to spraviť nejakto takto: `cout << char(66)`; vypíše B, keďže práve B má hodnotu 66. Funkcia `s.size()` vráti dĺžku `s`. Vtipnou záležitosťou sú “biele znaky” (*whitespace*

¹Ktorého?

²`string` v C++ je v skutočnosti `vector<char>` s nejakými dodatočnými funkciami.

characters) – to sú tie, ktoré nevidíme (medzera, tabulátor, enter). `cin >> s` štandardne načíta text iba po prvý biely znak. Na načítanie celého riadku do `stringu s` môžeme použiť `getline(cin,s)`.

V C++ vieme s `char` pracovať aj ako s číslom, napríklad `'B'+1` je `'C'`.

Nápoveda – C++, `cstdio`

Ukážeme si jednoduchý program, ktorý najprv načíta jedno číslo, potom načítava znaky až po koniec riadku (znak `'\n'`) a tento riadok vypíše. Následne načíta dve slová (slová sú oddelené bielymi znakmi) a vypíše ich na samostatné riadky.

```
#include <cstdio>
int main(){
    int cislo;
    char znak;
    char slovo[256];
    scanf("%d", &cislo);
    while(scanf("%c", &znak)>0 && znak!='\n') printf("%c", znak);
    printf("\n");
    scanf("%s", slovo);
    printf("%s\n", slovo);
    scanf("%s", slovo);
    printf("%s\n", slovo);
}
```

Odporúčame spustiť si tento program na ukážkovom vstupe zo zadania.

3. Zatúlané cukríky

kat. Z; 5 b za popis, 5 b za program

Kuko má rád sladkosti. Nepohrdne čokoládou, horalkou, ba ani čistým cukrom. No zo všetkého najradšej má rád cukríky. Takmer vždy má pri sebe otvorené vrecúško jeho obľúbených maškôt.

Jedného krásneho dňa šiel ako každý správny matfyzák na prednášku. A ako každý správny vedúci samozrejme na poslednú chvíľu. No ako obiehal Zergbota s kofolou, potkol sa a celé balenie sa mu rozsypalo po chodbe.

Rád by ich čo najviac pozbieral späť do sáčku, no nemôže sa veľmi zdržať, keďže už aj tak mešká na prednášku. Preto vás požiadal o pomoc.

Úloha

Chodbu na matfyzе si môžete predstaviť ako obdĺžnik $n \times m$ dláždený štvorcovými kachličkami. Keď Kuko prejde cez kachličku, zoberie všetky cukríky, ktoré sú na nej vysypané. Začína v ľavom hornom rohu chodby, prednášku má v pravom dolnom. Keďže sa na ňu ponáhľa, tak sa k nej chce v každom kroku posunúť bližšie. Môže teda chodiť iba doprava alebo dole.

Vašou úlohou je zistiť, koľko najviac cukríkov vie Kuko vyzbierať a ako sa má hýbať aby sa mu to podarilo.

Úloha sa testuje na piatich sádach vstupov. Pre vyriešenie prvých 2 sád nemusíte vypisovať Kukovu cestu, stačí zistiť počet cukríkov. Za takýto program dostanete 2 body.

Pre jednotlivé testovacie sady platia nasledovné obmedzenia:

Sada	Limity vstupov	Treba cestu
1	$n, m \leq 10$	Nie
2	$n, m \leq 1\,000$	Nie
3	$n, m \leq 10$	Áno
4	$n, m \leq 100$	Áno
5	$n, m \leq 1\,000$	Áno

Formát vstupu

Na prvom riadku vstupu dostanete čísla n a m – rozmery chodby. Nasleduje n riadkov, každý obsahuje m čísel oddelených medzerou – počty cukríkov na jednotlivých kachličkách. Tieto čísla sú v rozsahu od 1 po 10^6 .

Formát výstupu

Na prvý riadok vypíšte najväčší počet cukríkov, ktoré vie Kuko zozbierať. Na nasledujúci riadok vypíšte jeden reťazec bez medzier, reprezentujúci Kukovu trasu. Písmeno D označuje pohyb dole, písmeno R pohyb doprava. Ak existuje viacej optimálnych trás, vypíšte ľubovoľnú z nich.

Príklad

vstup	výstup
<pre>4 5 1 2 5 1 2 3 2 1 2 1 1 4 3 2 1 3 1 2 2 2</pre>	<pre>19 DRDRRDR</pre>

4. Zaspátý krtko

kat. Z a O; 7 b za popis, 8 b za program

Určite poznáte krtka Peťka. Celé dni vyspáva v svojej podzemnej nore a zobudí sa iba vtedy, keď sa ide najesť. Našťastie, býva priamo pod jablňovým sadom, stačí sa mu teda vykopať hore, pojesť čo najviac jabĺk a vrátiť sa do svojho pelechu, aby tam pokračoval vo svojej obľúbenej činnosti.

Jablňový sad je vlastne rad n stromov, pod ktorými sú napadané jablká. Peťko si vždy vyberie nejaký súvislý úsek jabloní, vykope sa na jeho začiatku, potom prejde až k jeho koncu a zakope sa späť pod zem. Popritom zje všetky jablká, na ktoré narazí.

Niektoré jablká však padli už veľmi dávno a preto zhnili. Peťko je však takmer úplne slepý a preto nevie rozlíšiť zhnité jablko od toho zdravého. Časom sa už zmieril s tým, že na to, aby sa dobre najedol, musí zjesť aj nejaké hnilé jablká. Snaží sa však, aby rozdiel medzi počtom zdravých a zhnitých jabĺk, ktoré zje, bol čo najväčší.

Po tomto príbehu je vám určite ľúto chudáka Peťka, s jeho ťažkým životným štýlom. Môžete mu však pomôcť. Stačí ak mu poradíte, ktorý úsek jabloní si má vybrať, aby sa čo najlepšie najedol.

Úloha

Na vstupe dostanete popis jablňového sadu ako postupnosť celých čísel. Pre každý strom budete vedieť, aký je rozdiel počtu zdravých a hnilých jabĺk pod týmto stromom. Keďže Peťko zje pod každým stromom všetky jablká, toto je naozaj jediná informácia, ktorú potrebujete.

Zistite, ktorý zo súvislých úsekov si má Peťko vybrať, aby bol rozdiel medzi počtom zdravých a zhnitých jabĺk, ktoré zje, najväčší možný.

Vstup

Prvý riadok vstupu obsahuje prirodzené číslo n ($1 \leq n \leq 1\,000\,000$) udávajúce počet stromov v sade.

Druhý riadok obsahuje n celých čísel p_i ($-1\,000\,000\,000 \leq p_i \leq 1\,000\,000\,000$) – rozdiely dobrých a zlých jabĺk pod jednotlivými stromami.

Výstup

Vypíšte jeden riadok a v ňom dve čísla a, b . Nech a ($1 \leq a \leq n$) je strom, pri ktorom sa má Peťko vykopať a b je počet stromov (vrátane stromu a), ktoré má prejsť smerom doprava. Ak existuje viacero riešení, vypíše ľubovoľné z nich.

Príklady

vstup	výstup
<pre>15 -1 -1 -1 1 2 3 4 5 -10 -10 -10 1 2 3 4</pre>	<pre>4 5</pre>
<p><i>Krtkovi sa oplatí vykopať pri strome číslo 4 s počtom jabĺk 1, spať potom 2, 3, 4 a 5 jabĺk pri nasledujúcich stromoch. Neoplatí sa mu ísť ďalej, takže sa zakope pri strome číslo 8.</i></p>	
<pre>5 -2 10 2 -3 4</pre>	<pre>2 4</pre>

vstup

3
1000000000 1000000000 1000000000

výstup

1 3

Dajte si pozor na veľkosť premennej, v ktorej si budete pamätať súčet!

vstup

5
-47 -42 -17 -13 -12

výstup

3 0

Môžte si všimnúť že správnych riešení je 5.

5. Otravné farmárčenie

kat. Z a O; 3 b za popis, 15 b za program

Predstavte si post- a zároveň pred- apokalyptický svet, v ktorom ľudstvo skoro vymrelo od hladu (preto post-) a čoskoro pravdepodobne aj vymrie (preto pred-). Hoci ste špičkoví programátori, musíte proti svojej vôli pracovať v poľnohospodárstve, aby ste mali čo jesť vy, vaše deti a iní príbuzní, nehovoriac o zvyšku ľudstva.

Kedže ľudstvo zanevrela na vedu, a tak odsúdilo Zem k úplnej záhube, jedinou nádejou na prežitie (okrem pestovania kukurice) je opustenie našej zničenej planéty. Aby sme Zem mohli opustiť, musíme najprv nájsť nejakú inú prívetivú planétu a chceme si byť istí, že bude skutočne obývateľná.

Problémom je, že všetky takéto planéty sú veľmi ďaleko a tak nám naše sondy vedľa posielajú len veľmi malé množstvo informácií. Mohli by sme sa uchýliť k tomu, že nám sondy budú posielajú len akési čísla a písmenká o prívetivosti planét, ale ako sa hovorí: „Lepšie raz vidieť, ako stokrát počuť.“

Čo ak na jednej planéte rastú kokosové palmy, šumí more a príjemne hreje neSlniečko a na druhej rastú kaktusy, šumí piesok a nepríjemne páli obrovská ohnivá guľa a my z oboch planét dostaneme rovnakú správu? To teda nie! Ak raz niekto bude posielajú takéto sondy, nedajbože aj s ľudskou posádkou, určite im treba dať aj foťáky a nejak vymyslieť, aby sa dali fotky posielajú.

Kedže vás úplne prestalo baviť pestovanie kukurice a zatúžili ste po troške dobrodružstva, na večernej prechádzke lesom ste sa snažili vlámať na oplotený súkromný pozemok. Za trest vás okamžite najali majiteľa pozemku a za úlohu vám dali vyriešiť problém vesmírnej komunikácie: Ako posielajú čo najkvalitnejšie fotografie v malom množstve dát?

Musíte to vymyslieť rýchlo. Majú vaše deti. A predsa len, je to zábavnejšie ako jazdenie na kombajne.

Úloha

Dostanete čiernobiely obrázok v textovom formáte veľkosti $w \times h$. Vašou úlohou je skomprimovať obrázok pomocou pár referenčných bodov tak, aby sa čo najviac podobal originálu.

Komprimovaný obrázok pozostáva z najviac $w + h$ bodov, pričom každý bod je popísaný súradnicami x, y a intenzitou I .

Pôvodný obrázok sa dekompresiou získa tak, že intenzita referenčných bodov zostane rovnaká a intenzita každého zvyšného pixelu sa vypočíta ako **vážený priemer** intenzít referenčných bodov, váhovaných prevrátenou hodnotou druhej mocniny vzdialenosti. Teda intenzitu pixela x, y z referenčných bodov $(x_1, y_1, I_1), \dots, (x_n, y_n, I_n)$ určíme ako:

$$I = \frac{\frac{I_1}{dx_1^2 + dy_1^2} + \frac{I_2}{dx_2^2 + dy_2^2} + \dots + \frac{I_n}{dx_n^2 + dy_n^2}}{\frac{1}{dx_1^2 + dy_1^2} + \frac{1}{dx_2^2 + dy_2^2} + \dots + \frac{1}{dx_n^2 + dy_n^2}}$$

kde $dx_i = (x - x_i)$ a $dy_i = (y - y_i)$.

Obrázky

Stiahni si [zip archív](#) z našej stránky. V ňom sa nachádzajú tri cvičné obrázky a 15 súťažných obrázkov. Každý obrázok sa nachádza vo formáte `.jpg` (klasický JPEG formát) a v takomto formáte `.in`:

Prvý riadok `.in` súboru obsahuje dve čísla w, h ($1 \leq w, h \leq 500$) oddelené medzerou – šírku a výšku obrázku.

Nasleduje $w \times h$ čísel p_i , ktoré môžu byť oddelené medzerou, alebo znakom nového riadku. Každé číslo $0 \leq p_i \leq 255$ vyjadruje intenzitu/odtieň sivej pixelu obrázku. Pixely sú zapísané v poradi zľava doprava, zhora dole, teda tak, ako by ste ich normálne čítali.³

0 je čierna, 255 je biela, čísla medzi určujú svetlejšie a tmavšie odtiene sivej.

³ p_i označuje intenzitu pixelu $[(i \bmod w) + 1, (i \operatorname{div} w) + 1]$, kde \bmod je zvyšok po delení, div je celočíselné delenie a $0 \leq i \leq w \times h - 1$

Odovzdávanie

Odovzdáva sa zip archív, ktorý obsahuje súbory 01.out, 02.out, atď až po 15.out. Nemusíte odovzdať všetky súbory, body dostanete za tie súbory, ktoré sa v archíve nachádzajú. (Ako obyčajne, do výsledkovky sa počíta len posledný odovzdaný zip archív, takže v ňom je odporúčané mať všetkých 15 súborov). Môžete do archívu pribalíť aj 00.sample.*.out súbory, ale tieto sú len cvičné a nedostanete za ne body.

Každý z odovzdaných súborov musí mať nasledovný formát:

Na prvom riadku sa nachádza číslo $n \leq w + h$ – počet bodov, ktorými chcete obrázok popísať.

Následne na n riadkoch sú po 3 čísla oddelené medzerami: x, y, I – súradnice a intenzity referenčných bodov. Zadané body musia ležať v obrázku, teda $1 \leq x \leq w, 1 \leq y \leq h$. Intenzity musia byť v rozmedzí od 0 po 255 vrátane.

Hodnotenie

Každý obrázok sa hodnotí samostatne. Na základe referenčných bodov obrázkov dekomprimujeme a porovnáme ho s originálom pixel po pixeli.

Pre obrázok vypočítame **strednú kvadratickú odchýlku** na základe rozdielov intenzít pixelov, teda sčítame druhé mocniny rozdielov intenzít a na záver výsledok predelíme počtom pixelov obrázku a odmocníme.

Za každý obrázok dostanete nejaký počet bodov od 0 po 1, podľa toho ako malá je vaša kvadratická odchýlka v porovnaní s referenčnou. Ak je vaša odchýlka o a referenčná odchýlka je r , tak počet bodov, ktoré dostanete bude $\max(0, \min(1, \frac{50+r-o}{50}))^2$, čo v praxi znamená, že plný počet bodov máte ak ste aspoň takí dobrí ako referenčná odchýlka a 0 bodov máte ak ste aspoň o 50 horší. Body, ktoré takto dostanete vám ešte nemusia ostať, zmenia sa, keď sa zmení referenčná odchýlka.

V tejto úlohe budete súťažiť proti sebe, takže referenčná odchýlka sa vypočíta na základe najlepšieho odovzdaného riešenia. Počas behu série budeme odchýlku aktualizovať približne raz za týždeň a po skončení série ju aktualizujeme znova a vypočítame finálne body.

Keďže neodovzdávate programy a nás zaujíma, ako úlohu riešite, môžete k úlohe odovzdať aj popis, kde stručne popíšete, ako ste úlohu riešili a priložíte zdrojové kódy programov, ktoré ste pri riešení použili. Za popis môžete získať 3 body. (Nemusíte odhadovať časovú zložitosť ani dokazovať správnosť riešenia, jednoducho napíšete, čo ste robili.)

Príklady s cvičnými obrázkami

vstup

```
7 7
255 255 255 255 255 255 255
255 255 0 255 0 255 255
255 255 255 255 255 255 255
255 0 255 255 255 0 255
255 255 0 255 0 255 255
255 255 0 0 0 255 255
255 255 255 255 255 255 255
```

výstup

```
9
1 1 255
7 1 255
3 2 0
5 2 0
4 3 255
3 5 0
5 5 0
1 7 255
7 7 255
```

Stredná kvadratická odchýlka tejto kompresie je 120,597746.

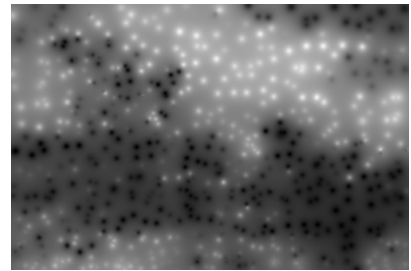
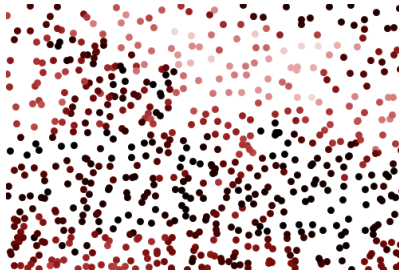
vstup

```
3 3
255 255 255
255 255 255
255 255 255
```

výstup

```
1
2 2 0
```

Úplne zlá kompresia má kvadratickú odchýlku 255. Ak by sme zmenili zadanú intenzitu na 255, kompresia by bola dokonalá a mala by odchýlku 0.



Na prvom obrázku je originálna [fotografia](#), na druhom môžete vidieť referenčné body, ktorými sa obrázok komprimuje. Na poslednom obrázku vidíte, ako by vyzeral obrázok po rekonštrukcii z referenčných bodov. Kvadratická odchýlka tejto kompresie je 49.131287. Vstupný súbor tohto obrázku si viete stiahnuť [tu](#), a výstup našej nedokonalkej kompresie [tu](#).

6. Overené informácie

kat. O; 10 b za popis, 10 b za program

KSP má tri písmena. P je 16. písmeno, S je 19. a K je 11. Keď odstránime dve jednotky z čísel 16 a 19 dostaneme 6 a 9. Koľko je 9? To je $3 \cdot 3$. Opakujem! Tri krát tri! TRI krát TRI. Koľko je 6? To je $3 + 3$. TRI + TRI! Náhoda? Myslím, že nie! Trojsten confirmed.

Niektoré veci proste nie sú náhodné. Čo sa stane, keď spojíte všetky “é” na tejto strane? Dostanete trojuholníky. Všetky trojuholníky! Každé Tri “é” tvoria trojuholník. To nie je náhoda! Trojuholníky nie sú náhoda. Prečo? Koľko majú strán? TRI! Keď veci tvoria veľa trojuholníkov, to nie je náhoda!

Úloha

Máte zadané body v rovine. Zistite, ako veľmi nenáhodné sú, teda vrcholy koľkých trojuholníkov tvoria.

Formát vstupu

Na prvom riadku vstupu je počet rôznych bodov n ($0 \leq n \leq 4000$).

Nasleduje n riadkov a na každom sú dve celé čísla x_i, y_i ($-100\,000 \leq x_i, y_i \leq 100\,000$), ktoré sú súradnicami daného bodu.

Formát výstupu

Vypíšte počet trojuholníkov s vrcholmi v daných bodoch.

Príklady

vstup

```
5
4 5
21 22
5 6
0 1
-5 -4
```

Tieto body sú jednoznačne náhodné.

vstup

```
5
0 0
3 3
3 0
0 3
1 -1
```

Toto naozaj nemôže byť náhoda.

výstup

0

výstup

10

7. Obsadenie toboganov plavčíkmi

kat. O; 15 b za popis, 10 b za program

Za mestom vznikol nový aquapark. Hore na kopci sú štartovacie stanovištia a dole sú veľké bazény. Zo stanovišť do bazénov vedú tobogany, na ktorých sa radostne šmýkajú deti. Občas, keď sa plavčík nepozerala, deti

skúšajú všelijaké vylomeniny, spúšťajú sa dolu hlavou, po trojiciach, alebo brzdia, keď nemajú. Inokedy ani plavčík nepomôže, niektoré deti robia hlúposti len vtedy, keď ich plavčík vidí lebo je to vraj viac cool a väčšia pocta.

Samozrejme, keď deti skúšajú nezbedy, stávajú sa úrazy a to nerobí aquaparku dobrú povest'. Prevádzkovatelia aquaparku by radi vedeli, aká je celková bezpečnosť aquaparku počas jednej sezóny.

Na každé stanovisko a ku každému bazénu môžeme aj nemusíme posadiť plavčíka. Rozloženie plavčikov sa každú chvíľu mení, až sa postupne vystriedajú úplne všetky možnosti. Bezpečnosť jedného toboganu závisí len od toho, či na jeho hornom a dolnom konci sedí plavčík. Bezpečnosť aquaparku v konkrétnom okamihu je súčin bezpečností všetkých toboganov v tom okamihu.

Úloha

Máte zadaný bipartitný graf,⁴ ktorý predstavuje náš aquapark. Hrany tohto grafu sú tobogany. Vrcholy tohto grafu sú bazény a štartovacie stanovištia a každý vrchol ofarbíme jednou z dvoch farieb – čiernou, ak vo vrchole sedí plavčík, a bielou, ak nesedí. Počet vrcholov označíme n a počet hrán m .

Pre každú hranu e vedúcu medzi vrcholmi u_e a v_e poznáme 4 hodnoty b_{e00} , b_{e01} , b_{e10} a b_{e11} , ktoré určujú, aká je bezpečnosť príslušnej hany podľa farieb jej koncových vrcholov.

- b_{e00} je bezpečnosť hrany e , ak sú oba vrcholy u_e , v_e biele.
- b_{e01} je bezpečnosť hrany e , ak je vrchol u_e biely a v_e čierny.
- b_{e10} je bezpečnosť hrany e , ak je vrchol u_e čierny a v_e biely.
- b_{e11} je bezpečnosť hrany e , ak sú oba vrcholy u_e , v_e čierne.

Všimnite si, že nehovoríme nič o tom, ktorý z vrcholov u , v je začiatok toboganu a ktorý koniec.

Pozrime sa na všetkých 2^n ofarbení grafu (pre všetky možnosti ako rozmiestniť plavčikov) a pre každé ofarbenie vypočítajme súčin bezpečností všetkých m hrán. Keď tieto súčiny sčítame, dostaneme celkovú bezpečnosť aquaparku počas sezóny.

Vypočítajte túto celkovú bezpečnosť modulo $10^9 + 7$.

Formát vstupu

Na prvom riadku vstupu sú čísla n a m , ($2 \leq n \leq 40$, $1 \leq m \leq 100$). Vrcholy očísľujeme postupne 1 až n .

Nasleduje m riadkov a na každom je 6 čísel $u_e, v_e, b_{e00}, b_{e01}, b_{e10}, b_{e11}$ popisujúcich jednu hranu ($1 \leq u_e, v_e \leq n$, $0 \leq b_{eij} \leq 10^9$).

Čiastočné body sa budú dať získať aj za riešenia, ktoré zvládajú len malé n, m , prípadne malé hodnoty b . Budú aj vstupy, v ktorých je celková bezpečnosť pomerne malá.

Formát výstupu

Vypíšte jedno celé číslo – celkovú bezpečnosť lunaparku počas sezóny modulo 1 000 000 007.

Príklady

vstup	výstup
<pre>2 1 1 2 1 2 3 4</pre>	<pre>10</pre>
vstup	výstup
<pre>3 2 1 2 1 0 0 1 2 3 1 0 0 1</pre>	<pre>2</pre>

8. Okázalá svadba

kat. O; 15 b za popis, 10 b za program

Usáma a Maru sa rozhodli, že spravia tú najokázalejšiu svadbu, akú kedy Trojsten videl. Rozhodli sa preto, že každý hosť dostane ručne šitého motýlika alebo mašľu do vlasov. Výhodou je, že pri objednávaní to netreba rozlišovať, lebo je to vlastne to isté, len inak nazvané. V krajčírskkej firme J&P⁵ si teda objednali zákazku na n motýlikov.

⁴To je taký graf, ktorého vrcholy sa dajú rozdeliť na dve časti (partície) tak, aby v rámci jednej časti nevedli žiadne hrany. V našom prípade jednu partíciu tvoria štarty toboganov a druhú partíciu bazény. Tobogan vždy vedie medzi bazénom a stanovištom. Tiež sa dá všimnúť, že bipartitné grafy sú práve tie grafy, ktoré neobsahujú cykly nepárnej dĺžky.

⁵Jano a podšívky.

Jano je veľký motýlikový umelec a na tejto zákazke sa vybláznil, preto je každý motýlik iný a originálny. Ako však vieme, nie každý originálny nápad dopadne úspešne a teda nie všetky motýliky vyzerajú rovnako dobre. Každému motýliku preto Usáma a Maru priradili číslo zodpovedajúce jeho kráse.

Situácia sa však skomplikovala. Nie sú si totiž istí, koľko pozvaných hostí naozaj príde. Chceli sa preto s Janom dohodnúť, že keď zistia počet hostí k , ktorí prídu na svadbu, zoberú si k najkrajších motýlikov. To im však Jano zakázal, zoradil motýliky podľa poradia ušitia a povedal, že si budú môcť vybrať iba súvislý úsek dĺžky k .

V tom okamihu sa rozpútala hádka⁶. Usáma chcel, ako správny egoista, získať najkrajší motýlik pre seba a chcel preto úsek s čo najväčším maximom krásy. Maru je ale štedrá osoba a páči sa jej rozmanitosť, preto chcela maximalizovať bitový OR krás všetkých motýlikov, ktoré budú vo vybranom úseku.

Nakoniec sa dohodli na kompromise⁷ a rozhodli sa maximalizovať súčet maximálneho prvku a bitového ORu všetkých krás v úseku. Stále však nevedia, koľko hostí príde na svadbu, preto by chceli vedieť dopredu, ktorý úsek vybrať pre všetky možné k .

Úloha

Na vstupe dostanete postupnosť kladných celých čísel dĺžky n . Vašou úlohou bude pre každé k od 1 po n nájsť taký súvislý úsek dĺžky k v tejto postupnosti, že súčet maxima v tomto úseku a bitového ORu prvkov tohoto úseku je najväčší možný.

Formát vstupu

Na prvom riadku sa nachádza číslo n ($1 \leq n \leq 100\,000$), ktoré označuje počet vyrobených motýlikov. Nasleduje n čísel, ktoré predstavujú krásy jednotlivých motýlikov v poradí, v akom boli vytvorené. Krásy motýlikov sú kladné celé čísla neprevyšujúce 10^9 .

Formát výstupu

Vypíšete n čísel na samostatné riadky, kde k -te z nich predstavuje najväčší možný súčet požadovaných vlastností pre úseky dĺžky k .

Príklad

vstup	výstup
3 1 0 2	4 4 5

Zadania kategórie T

Nezabudnite, že môžete riešiť aj kategóriu T (je trocha ťažšia ako kategória O, ale mnohí z vás ju určite zvládnu).

⁶V skutočnosti to tak nebolo. Maru rozhodla, ktorý úsek zoberú a Usáma nemal na výber. Takýto scenár by však nevytvoril zaujímavé zadanie.

⁷Čo je v manželstve veľmi dôležité.



Najúspešnejších riešiteľov letnej série KSP čaká úžasná odmena!

Spoločnosť SAIC pozve troch najlepších riešiteľov na pobyt v San Diegu. Viac informácií nájdete na stránke ksp.sk/san_diego.

