

Na řešení úloh máte 4,5 hodiny čistého času. Při soutěži je zakázáno používat jakékoliv pomůcky kromě psacích potřeb a přiděleného počítače (tzn. knihy, kalkulačky, mobily, apod.).

Řešením každého příkladu je zdrojový kód programu zapsaný v programovacím jazyce Pascal, C nebo C++. Řešení odevzdáváte pomocí soutěžního systému CMS, který ho automaticky otestuje na připravených sadách testovacích dat. Za každou úlohu můžete získat až 10 bodů. Detaily hodnocení naleznete v letáku s popisem prostředí. Podrobnější informace o testovacích datech najdete na konci zadání každé úlohy.

P-III-4 Laser

Program: laser.pas / laser.c / laser.cpp
Vstup a výstup: standardní vstup a výstup
Časový limit: najdete v soutěžním systému
Paměťový limit: najdete v soutěžním systému

Na povrchu obdélníkového stolu je nakreslena čtvercová síť. Na některých políčkách sítě, včetně téměř všech políček na obvodu stolu, jsou překážky. Na obvodu stolu jsou jen dvě výjimky. Jednou z nich je políčko, na němž je umístěn laser, namířený dovnitř stolu. Druhou výjimkou je políčko, kde je fotoreceptor: zařízení, které umí rozpoznat, když je zasaženo laserovým paprskem.

Na některých políčkách stolu jsou postaveny skleněné kostky, kterými nesmíte pohnout. Laserový paprsek prochází kostkami bez jakýchkoliv efektů.

Na volná políčka stolu můžete umísťovat zrcadla. Každé zrcadlo je oboustranné (tj. obě jeho strany odrážejí) a musí být umístěno na úhlopříčce políčka. Na volné políčko tedy můžeme umístit zrcadlo dvěma způsoby. Laserový paprsek při dopadu na zrcadlo změni svůj směr o 90 stupňů.

Soutěžní úloha

Je dána bitmapa představující současný stav stolu. Zjistěte, zda je možné vhodným přidáním zrcadel přivést laserový paprsek od laseru k fotoreceptoru. Pokud ano, určete nejmenší počet zrcadel, který je k tomu zapotřebí, a naleznete jedno vyhovující rozmístění optimálního počtu zrcadel. (Každé použité zrcadlo počítáme jen jednou, i kdyby se od něj laserový paprsek postupně odrazil dvakrát – od každé jeho strany jednou.)

Formát vstupu a výstupu

Na prvním řádku vstupu jsou rozměry stolu r a s . Zbytek vstupu tvoří bitmapa popisující stůl: r řádků a na každém z nich s znaků. Znak ‘.’ (tečka) představuje

P-III-5 Graffiti

<i>Program:</i>	<code>graffiti.pas / graffiti.c / graffiti.cpp</code>
<i>Vstup a výstup:</i>	standardní vstup a výstup
<i>Časový limit:</i>	najdete v soutěžním systému
<i>Paměťový limit:</i>	najdete v soutěžním systému

Podél nejmenované státní hranice kdosi nedávno postavil dlouhatánskou zeď. Skládá se z p panelů, které si pro účely této úlohy očísľujeme postupně od 1 do p . Nově postavená zeď byla krásně čistá. V budoucnosti ale očekáváme, že budou probíhat dva typy událostí:

- Občas přijde samozvaný umělec s barvami a bude chtít pomalovat nějaký souvislý úsek čistých panelů.
- Občas přijde pořádková služba a vyčistí nějaký souvislý úsek panelů (ať už čistých, nebo pomalovaných).

Soutěžní úloha

Vaším úkolem je napsat program, který bude takovéto události efektivně zpracovávat.

O pořádkové službě se vždy dozvíte přesný úsek panelů, který vyčistí.

O umělci se dozvíte jenom délku úseku čistých panelů, které chce pomalovat. O umělcích navíc víme, že všichni bydlí u začátku zdi a jsou hodně líní. Má-li umělec na výběr více různých dostatečně dlouhých čistých úseků zdi, vždy si vybere ten, který začíná panelem s nejmenším číslem.

Formát vstupu a výstupu

Na prvním řádku vstupu jsou dvě kladná celá čísla: počet panelů p a počet událostí u .

Zbytek vstupu tvoří u řádků, každý popisuje jednu událost v chronologickém pořadí. Příchod umělce bude popsán řádkem ve tvaru „ $U d$ “, kde d je délka úseku, který chce pomalovat. Příchod pořádkové služby bude popsán řádkem ve tvaru „ $P x y$ “, kde $x \leq y$ jsou čísla prvního a posledního panelu v právě vyčištěném úseku.

Pro každý příchod umělce vypište jeden řádek s jedním číslem: číslo panelu, na němž začíná úsek, který umělec pomaluje.

Pokud se někdy stane, že na zdi není nikde tak dlouhý čistý úsek, jaký by umělec potřeboval, zklamaný umělec se vrátí domů a nic nepomaluje. V takovém případě vypište číslo -1 .

Omezení a hodnocení

Je připraveno 5 sad testovacích vstupů, za každou můžete získat 2 body.

- V první sadě platí $p \leq 100$ a $u \leq 1000$.
- Ve druhé sadě platí $p, u \leq 50\,000$.
- Ve třetí sadě platí $p, u \leq 150\,000$.
- Ve čtvrté a v páté sadě platí $p \leq 500\,000$ a $u \leq 300\,000$. Ve čtvrté sadě navíc pro každého umělce platí $d = 1$.
- Ve všech sadách v každé události platí $1 \leq d \leq 10^9$ a $1 \leq x \leq y \leq p$.

Příklad

<i>Vstup:</i>	<i>Výstup:</i>
10 6	1
U 6	-1
P 2 4	7
U 5	2
U 4	-1
U 2	
U 47	

První umělec pomaluje panely 1 až 6. Pořádková služba vyčistí panely 2 až 4. Druhý umělec by chtěl pomalovat pět panelů, nikde ale není pět čistých panelů vedle sebe. Třetí umělec pomaluje panely 7 až 10 – to je v dané chvíli jediná čtveřice čistých panelů vedle sebe. Čtvrtý umělec pomaluje panely 2 a 3. Pátý umělec by toho chtěl pomalovat více, než kolik měří celá zeď, má tedy smůlu.

P-III-6 Korálky

<i>Program:</i>	<code>koralky.pas / koralky.c / koralky.cpp</code>
<i>Vstup a výstup:</i>	standardní vstup a výstup
<i>Časový limit:</i>	najdete v soutěžním systému
<i>Paměťový limit:</i>	najdete v soutěžním systému

Ještě nedávno měla Natálka krásný náhrdelník, na kterém bylo n korálků. Korálky měly různé odstíny růžové barvy, postupně přecházely z tmavé do světlé. Náhrdelník se ale přetrhl a korálky se rozkutálely po zemi.

Natálka teď drží prázdnou šňůrku a Petr (který se na přetržení náhrdelníku maličko podílel) klečí na zemi, sbírá korálky a jeden po druhém je Natálce podává. Ta může každý korálek buď navléknout na šňůrku (a to z libovolného konce), nebo ho může odložit do krabice s šitím.

Natálku při sbírání korálků napadla následující úloha: Co kdyby korálky zkusila navlékat tak, aby nakonec byly na šňůrce seřazené od nejtmaššího po nejsvětější? Možná se jí nepodaří takto navléknout všechny korálky, ale určitě bude zábavné zkusit jich navléknout co nejvíce.

Soutěžní úloha

Na vstupu je dána posloupnost f_1, \dots, f_n : barvy korálků v tom pořadí, v jakém je Petr podával Natálce. Všechna f_i jsou *navzájem různá* přirozená čísla. Čím větší číslo, tím světlejší korálek představuje.

Pro každý korálek si Natálka vybrala jednu ze tří možností: buď ho navlékla na levý konec šňůrky, nebo ho navlékla na pravý konec šňůrky, nebo ho odložila pryč. Víme, že na konci celého procesu měla šňůrku, na níž zleva doprava barva korálků přecházela z tmavé do světlé – tedy jim odpovídající čísla rostla.

Napište program, který určí, kolik nejvýše korálků mohla mít nakonec Natálka na šňůrce.

Formát vstupu a výstupu

Na prvním řádku vstupu je kladné celé číslo n . Na druhém řádku je n různých kladných celých čísel f_1, \dots, f_n .

Vypište jeden řádek s jedním číslem: maximální počet korálků, které mohly skončit na šňůrce.

Omezení a hodnocení

Je připraveno 10 sad testovacích vstupů, za každou můžete získat 1 bod. Ve všech sadách pro každé i platí $1 \leq f_i \leq 10^9$.

Jednotlivé sady mají různou maximální hodnotu n : 6, 12, 20, 60, 100, 5 000, 17 000, 150 000, 250 000 a 500 000.

Příklady

Vstup:

5
30 10 50 20 40

Výstup:

3

Jedno optimální řešení: Korálek 30 navleče na šňůrku (je jedno z které strany), korálek 10 nepoužije, korálek 50 navleče zprava, korálek 20 navleče zleva a korálek 40 nepoužije.

Vstup:

6
10 20 9 21 8 22

Výstup:

6

Všechny korálky můžeme navléknout na šňůrku.

Vstup:

7
1 7 4 3 5 2 6

Výstup:

5

Zde je optimálním řešením nepoužít první dva korálky a z ostatních sestavit na šňůrce posloupnost 2, 3, 4, 5, 6.