Lăcusta

O lăcustă sare în fiecare zi a săptămânii **n** metri, iar în zilele de sâmbătă și duminică sare câte **m** metri. Știind că numărătoarea începe într-o zi de luni să se calculeze cât a sărit lăcusta după **z** zile.

Date de intrare: Fișierul **lacusta.in** conține pe prima linie 3 numere naturale **n**, **m** și **z** cu semnificația din enunț.

Date de ieșie: Fișierul **lacusta.out** conține un număr natural reprezentând distanța parcursă de lăcustă după **z** zile.

Restricții și precizări:

- $1 \leq$ **n, z, m** $\leq 1\,000$.

Exemplu:

lacusta.in	**lacusta.out**	**Explicație**
5 2 8	34	

Bomboane

Lucy lucrează **n** ore într-o fabrică de bomboane. Bomboanele care vin pe banda rulantă trebuie împachetate și puse în cutii. La început, pe bandă, sosesc câte **k** bomboane în fiecare oră. Din **p** în **p** ore bomboanele sunt transportate în magazine astfel: dacă în cutii sunt un număr par de bomboane jumătate din ele vor fi transportate, iar dacă în cutii sunt un număr impar de bomboane Lucy primește cadou o bomboană, iar din ceea ce a rămas jumătate sunt transportate. După un transport numărul de bomboane care vin pe bandă crește cu **1**.

Cerință: cunoscând **n** (numărul de ore pe care le lucrează Lucy), **k** (numărul de bomboane pe oră care vin inițial pe banda rulantă) și **p** (intervalul de timp la care bomboanele sunt transportate) scrieți un program care calculează câte bomboane se găsesc în cutii după **n** ore.

Date de intrare: fișierul **bomboane.in** conține 3 numere naturale **n, k** și **p**, separate printr-un spațiu, cu semnificația din enunț.

Date de ieșire: fișierul **bomboane.out** va conține un număr natural, reprezentând numărul de bomboane se care găsesc în cutii după **n** ore.

Restricții și precizări:

- $1 \leq \mathbf{p} \leq \mathbf{n} \leq 100\,000$;
- $1 \leq \mathbf{k} \leq 2\,000$;
- Lucy lucrează foarte repede, așa că timpul de împachetare este neglijabil;

Exemplu:

bomboane.in	bomboane.out	Explicație
8 4 3	22	n=8, k=4, p=3 (la început vin 4 bomboane/oră, care sunt transportate din 3 în 3 ore). Ora 1: în cutii sunt 4 bomboane Ora 2: în cutii sunt 8 bomboane Ora 3: în cutii sunt 12 bomboane. Jumătate sunt transportate, rămân 6. De acum vin câte 5 bomboane/oră. Ora 4: în cutii sunt 11 bomboane. Ora 5: în cutii sunt 16 bomboane. Ora 6: în cutii sunt 21 de bomboane. Lucy primește una și jumătate sunt transportate. Rămân 10 bomboane. De acum vin câte 6 bomboane/oră. Ora 7: În cutii sunt 16 bomboane. Ora 8: în cutii sunt 22 de bomboane.

Timp de execuție/test: 1s.

Memorie disponibilă: 4MB din care 2MB pentru stivă.

Vampiri

Războiul între vampiri nu este un simplu joc. Fiecare clan are o strategie a sa, ținută bine ascunsă de ochii curioșilor. Fiind susținător înverșunat al clanului Delany, norocul îi surâde lui Bonnie: Diego, liderul clanului, îi dă o misiune care va avea rol decisiv în rezultatul acestui război. Bonnie trebuie să urmărească mișcările vampirilor clanului advers, și să îi spună lui Diego poziția în care a ajuns clanul McCuloth la finalul seriei de mișcări, ajutându-l astfel pe acesta să le descopere strategia.

Bonnie a descoperit că cei din clanul McCuloth se deplasează în linie dreaptă, pornind din poziția **0**, efectuând două tipuri de mișcări:

- Mișcări cu **x** pași la stânga poziției curente, codificate prin litera **S** urmată de numărul **x**. De exemplu, dacă vampirii vor efectua **3** pași la stânga poziției curente, codificarea acestei mișcări va fi: **S 3**.

- Mișcări cu **x** pași la dreapta, codificate prin litera **D** urmată de numărul **x**. De exemplu, dacă vampirii vor efectua **5** pași la dreapta poziției curente, codificarea acestei mișcări va fi: **D 5**.

Cerință. Dându-se numărul **n** de mutări și un șir succesiv de **n** mișcări codificate ca în exemplele de mai sus, să se determine poziția finală în care au ajuns vampirii din clanul McCuloth după executarea celor **n** mișcări. Se știe că poziția de start a vampirilor este egală cu **0**.

Date de intrare. Fișierul **vampiri.in** conține pe prima linie un număr natural **n**. Pe următoarele **n** linii se găsesc **2** valori naturale pe fiecare rând, un caracter și o valoare naturală, reprezentând mișcările codificate.
Date de ieşire: fişierul de ieşire **vampiri.out** va conţine pe prima linie codificarea poziţiei în care se vor afla la sfârşit. Această codificare este făcută tot în raport cu 0, după aceeaşi regulă : dacă la final vampirii se află în stânga cu x paşi, atunci se va scrie Sx; dacă se ajunge în dreapta cu x paşi, se va scrie Dx. Dacă poziția inițială 0 coincide cu poziția finală, atunci fișierul va conține pe prima linie valoarea 0.

Restricţii şi precizări
- $1 \leq n \leq 100$;
- $0 \leq x \leq 9$.

Exemplu:

vampiri.in	**vampiri.out**	***Explicație***
4 D 3 S 2 D 5 S 9	S3	Se va rezolva numai punctual b) din cerință. Vampirii se află inițial pe poziția 0 și efectueaza 4 mutări. Mai intâi, 3 pași la dreapta, ajungând cu 3 paşi la dreapta lui 0; 2 pași la stânga, ajungând cu 1 pas la dreapta lui 0; 5 pași la dreapta, ajungând în poziția D6 și, în final, 9 pași la stânga, ajungând la 3 paşi în stânga faţă de poziţia de start 0 (S3).

Poliția

Departamentul de poliție din orașul tău tocmai și-a început activitatea. Inițial, nu au forță de muncă. Așa că, au început să angajeze noi recruți.

Între timp, crimele continuă să se producă în oraș. Un membru al poliției poate investiga o singură infracțiune în timpul carierei sale. Dacă nu există niciun ofițer de poliție liber (nu este ocupat cu crima) în timpul producerii unei infracțiuni, aceasta va rămâne necercetata.

Cerință Având în vedere ordinea cronologică a crimelor și a angajărilor, găsiți numărul de infracțiuni care nu vor fi cercetate.

Date de intrare Prima linie de intrare va conține un număr întreg **n** ($1 \leq n \leq 10^5$), numărul de evenimente. Următoarea linie va conține **n** numere întregi separate prin spațiu. Dacă numărul întreg este **-1** înseamnă că a avut loc o crimă. În caz contrar, numărul întreg va fi pozitiv, numărul de ofițeri recrutați împreună la acel moment. Nu vor fi recrutați mai mult de **10** ofițeri simultan.

Date de ieşire: Tipăriți un singur număr întreg, numărul de infracțiuni care nu vor fi cercetate.

intrare	**ieşire**	
8 1 -1 1 -1 -1 1 1 1	1	1. În primul rând este angajată o persoană. 2. apare infracțiunea, ultima persoană angajată va cerceta această infracțiune. 3. Mai este angajată o persoană. 4. Mai apare o infracțiune, ultima persoană angajată va cerceta această infracțiune. 5. Apare crima. Nu există niciun polițist liber în acel moment, așa că această crimă nu va fi tratată. 6. Mai este angajată o persoană. 7. Mai este angajată o persoană. 8. Mai este angajată o persoană.
11 -1 -1 2 -1 -1 -1 -1 -1 -1 -1 -1	8	

CANGURUL

A fost odată ca niciodată, a fost un cangur care creştea ca Făt Frumos din poveste, într-un an precum alţii în zece. Într-o zi a început să facă sărituri după cum urmează:

- În prima zi a sărit **k** metri.
- A doua zi a sărit în plus faţă de prima zi cu de **p** ori lungimea saltului din prima zi.
- A treia zi a sărit în plus faţă de prima zi cu de **p** ori lungimea saltului din a doua zi.

La fel a sărit şi în zilele următoare până în ziua cu numărul **n.**

Lungimea totală pe care cangurul a reuşit să o sară în cele **n** zile este dată de suma lungimilor săriturilor din cele **n** zile.

Cerinţă : scrieţi un program **cangur.cpp** care calculează câţi metri a sărit cangurul, în total, în cele **n** zile.

Date de intrare : se citesc valorile **k, p** şi **n** în această ordine, cu semnificația din enunț.

Date de ieşire : se afiseaza o singură valoare naturală reprezentând câţi metri a sărit cangurul, în total, după cele **n** zile.

Restricţii şi precizări

- **n, p** şi **k** sunt numere naturale
- $0 < n \leq 100$
- $0 < k, p \leq 10$

Exemplu

intrare	iesire	Explicaţie
7 10 3	861	I zi: 7 metri II zi: 7+70= 77 metri III zi: 7+770=777 metri 7 metri+ 77 metri+ 777 metri= 861 metri

Timp maxim de execuţie/test: 1 secundă.