PROBLEMA 1 **100 puncte**

NR

Un număr natural nenul V care se plictisea singur, și-a căutat în prima zi cel mai mare divizor al său mai mic decât el și l-a scăzut din valoarea sa. Numărul rămas, plictisit și el, și-a căutat a doua zi cel mai mare divizor al său mai mic decât el și l-a scăzut din valoarea sa. Și așa mai departe până ce numărul rămas a ajuns 1, cand s-a declarat fericit și nu a mai căutat nimic.
Dacă numărul inițial V devine fericit după **cel mult** Z zile, spunem că el este Z-fericibil.
De exemplu, numărul 10 devine fericit după 4 zile (zi_1: 10-5=5, zi_2: 5-1=4; zi_3: 4-2=2; zi_4: 2-1=1), deci el este 4-fericibil, 5-fericibil, 6-fericibil etc. dar nu este 1-fericibil, nici 2-fericibil, nici 3-fericibil.

Cerinţă:
Pentru o valoare Z dată și un șir de N valori, V_1, V_2, ..., V_N, se cere să se determine câte dintre valorile V_1, V_2, ..., V_N sunt Z-fericibile.

Date de intrare:
Din fișierul ***nr.in*** se citesc numere naturale N și Z, despărțite printr-un spațiu, de pe prima linie, și apoi N numere naturale V_1, V_2, ..., V_N despărțite prin câte un spațiu, de pe linia a doua a fișierului.

Date de ieşire:
În fișierul ***nr.out*** se va scrie un număr F, reprezentând numărul de valori din șirul V_1, V_2, ..., V_N care sunt Z-fericibile.

Restricţii și precizări:
- $1 \leq N \leq 100000$; $1 \leq Z \leq 30$
- Pentru 80% din teste, $1 \leq V_i \leq 2000000000$, pentru orice **i** între 1 și N
- Pentru 20% din teste, $1 \leq V_i \leq 1000000000000000000$, pentru orice **i** între 1 și N
- Numerele V_1, V_2, ..., V_N nu sunt neapărat distincte

Exemple:

nr.in	nr.out	Explicații
3 4 10 28 2	2	Dintre cele 3 numere date, doar două sunt 4-fericibile: – 10 devine fericit după 4 zile (vezi exemplul din enunț), deci este 4-fericibil; – 28 devine fericit după 6 zile (28-14=14; 14-7=7;7-1=6; 6-3=3; 3-1=2; 2-1=1), deci nu este 4-fericibil; - 2 devine fericit după o zi (2-1=1), deci este 4-fericibil.
4 2 7 28 6 31	0	Dintre cele 4 numere date, niciunul nu este 2-fericibil: – 7 devine fericit după 4 zile (7-1=6;6-3=3;3-1=2;2-1=1), deci nu este 2-fericibil; – 28 devine fericit după 6 zile (vezi exemplul anterior), deci nu este 2-fericibil; – 6 devine fericit după 3 zile (6-3=3; 3-1=2; 2-1=1), deci nu este 2-fericibil; – 31 devine fericit după 7 zile (31-1=30; 30-15=15; 15-5=10; 10-

		5=5; 5-1=4; 4-2=2; 2-1=1), deci nu este 2=fericibil.

Timp maxim de executare/test: 0.2 sec
Memorie totală: 2MB din care 2MB stiva
Dimensiunea maximă a sursei: 5KB

PROBLEMA 2 **100 puncte**

COVOR

Iustin este un băiețel căruia îi plac numerele. A învățat la școală adunarea și scăderea și singur înmulțirea. Acum însă vrea să știe mai multe despre împărțire și a observat că există numere care nu se împart decât la 1 și la ele însele și a aflat că se numesc numere prime. Mama sa, ca să îl ajute să memoreze cât mai multe astfel de numere, a țesut un covor pentru camera lui, format din *n x n* pătrate, pe fiecare pătrat fiind desenat un număr prim. Pentru că Iustin este încă mic, mama s-a gândit să scrie de mai multe ori numerele prime impare cele mai mici și de mai puține ori pe cele mari. Ajut-o să formeze un covor conform cerinței.

Cerinţă:
Să se afișeze în fișierul *covor.out* modelul covorului, știind că:

- chenarul format din linia *1,* coloana *1,* linia *n* și coloana *n* are numai elemente egale cu primul număr prim impar,
- chenarul format din *linia 2,* coloana *n-1,* linia *n-1 și coloana 2* are numai elemente egale cu al doilea număr prim
- etc.
- în centrul covorului se află al *n/2+1*-lea număr prim

Exemplu: pentru *n=7* se obține:

3	3	3	3	3	3	3
3	5	5	5	5	5	3
3	5	7	7	7	5	3
3	5	7	11	7	5	3
3	5	7	7	7	5	3
3	5	5	5	5	5	3
3	3	3	3	3	3	3

Date de intrare:
Pe prima linie a fișierului ***covor.in*** se află un număr natural impar *n*

Date de ieşire:
În fișierul ***covor.out*** se afișează modelul cerut.

Secţiunea 5-6 avansaţi

Restricţii şi precizări:
- numărul natural n este impar, 1≤n≤835
- al 417-lea număr prim impar are valoarea 2887

Exemplu:

covor.in	**covor.out**	**Explicaţie**
7	3 3 3 3 3 3 3 3 5 5 5 5 5 3 3 5 7 7 7 5 3 3 5 7 11 7 5 3 3 5 7 7 7 5 3 3 5 5 5 5 5 3 3 3 3 3 3 3 3	Se construiesc chenarele formate din numere prime impare în ordine crescătoare; ultimul chenar conține un pătrat cu singur număr, în exemplul acesta 11

Timp maxim de execuție: 0,2 secunde/test.
Memorie totală disponibilă 5 MB, din care **2** MB pentru stivă
Dimensiunea maximă a sursei: **5** KB.

PROBLEMA 3 **100 puncte**

CULORI

Alchimistul Algorel doreşte să experimenteze nuanţe noi pornind de la cutii cu vopsea găsite în garajul tatălui său. Astfel, din cutiile cu vopsele în culorile de bază, prin diluare în proporţii numai de el cunoscute, obţine un set de **n** cutii cu vopsea din care va crea nuanţe noi prin amestecare. Pe fiecare cutie a scris un număr **c** care reprezintă codul culorii.
El adaugă într-o eprubetă pe rând câte o picătură din fiecare cutie. După mai multe încercări el observă că în anumite combinaţii, culorile se anulează între ele, revenind la culoarea de înainte de a adăuga „combinaţia”. De aceea el decide ca aceste culori să nu mai apară în formula finală (pentru a nu face risipă de vopsea). Tot în urma acestor încercări a observat că acele culori care se combină anulându-se una pe cealaltă, au următoarele proprietăţi:
- sunt trei culori adăugate una după cealaltă;
- culorile c_i, c_{i+1}, c_{i+2} se anulează dacă între *puterile* lor este relaţia: p_{i+1} este medie aritmetică între p_i şi p_{i+2}.

Puterea unei culori este o cifră **nenulă** obţinută astfel: se calculează suma cifrelor care alcătuiesc codul culorii, iar dacă această sumă nu este o cifră atunci se repetă procedeul până când suma respectivă este o cifră. De exemplu, dacă pe o cutie este scris codul 8146 atunci culoarea are puterea 1, deoarece 8+1+4+6 este 19, care nu este cifră şi reluăm: pentru 19 suma cifrelor este 10, care nu este cifra şi reluăm: pentru 10 suma cifrelor este 1. Aşadar puterea culorii 8146 este 1

La întâlnirea primei combinaţii de culori care se anulează între ele, acestea sunt eliminate din formulă şi se reia experimentul de la început, cu formula modificată, până când formula nu mai conţine astfel de combinaţii.

Cerinţă:
Cunoscând numărul de cutii şi codul scris pe fiecare cutie, scrieţi un program care să afişeze codurile culorilor din formula nuanţei noi descoperită de Algorel. Dacă prin combinare, formula finală nu mai conţine nici un cod, atunci se va afişa mesajul MAI INCEARCA.

Date de intrare:
Fişierul ***culori.in*** are următoarea structură:
- pe prima linie un număr natural **n** reprezentând numărul de cutii cu vopsea;
- pe a doua linie cele **n** numere scrise de Algorel pe fiecare cutie.

Date de ieşire:
Fişierul ***culori.out*** conţine pe o linie, separate prin spaţiu, codurile culorilor care rămân în formula finală.

Restricţii şi precizări:
- 3 < = n < = 100000;
- codul unei culori este un număr natural nenul cu cel mult 9 cifre.

Exemple:

culori.in	**culori.out**	**Explicaţii**
7 1111 21 24 405 5189 308 9125	1111 5189 308 9125	Culorile introduse au, în ordine, următoarele puteri: 4, 3, 6, 9, 5, 2, 8. Prima combinaţie de culori care se anulează între ele este formată din codurile 21, 24, 405, deoarece p(24) este media aritmetică între p(21) şi p(405). Aşadar experimentul se reia de la început, cu formula formată din culorile cu codurile 1111, 5189, 308, 9125, între care nu mai întâlnim combinaţii care se anulează. Aşadar acestea sunt culorile din formula finală.
6 100 23 211 813 6626 39	MAI INCEARCA	Prima combinaţie de culori care se anulează între ele este 23, 211, 813. Acestea sunt şterse din formulă şi experimentul se reia de la început cu formula formată din culorile cu codurile 100, 6626 şi 39. Pentru aceste trei culori se observă procesul de anulare, iar la reluarea experimentului observăm că formula nu mai conţine nici un cod.

Timp maxim de execuţie: **0,2** secunde/test.
Memorie totală disponibilă 2 MB, din care 1 MB pentru stivă
Dimensiunea maximă a sursei: **5** KB.

PROBLEMA 4 **100 puncte**

GREA

Empowerboss are foarte multă treabă, așa că s-a gândit să vă ocupe timpul cu o problemă foarte grea.
Acesta vă dă un şir cu T numere naturale. Pentru fiecare număr din şir, să-l notăm cu A trebuie să găsiți cel mai mare număr K cu proprietatea că există un șir de K numere naturale B_1,B_2... ,B_k, nu neapărat distincte, astfel încât: $(B_1 + 1)(B_2 + 1)...(B_K + 1) = A$

Cerința:
Arătați-i lui Empowerboss că problema nu e suficient de grea pentru voi, găsind numărul K cerut într-un timp cât mai scurt, pentru fiecare din cele T numere.

Date de intrare:
În fișierul ***grea.in***, pe prima linie se află numărul T, iar pe următoarele T linii câte unul din numerele date de Empowerboss.

Date de ieșire:
Fișierul ***grea.out***, va conţine pe fiecare linie un număr K, reprezentând numărul maxim de termeni pe care îi poate avea șirul B, astfel încât să respecte proprietatea cerută. Prima linie reprezintă răspunsul pentru primul număr, a doua pentru cel de-al doilea ... şamd.

Restricții si precizări:
$1 \leq T \leq 500$
$2 \leq A \leq 2.000.000.000$

Exemplu:

grea.in	grea.out	Explicaţie
1 4	2	Ne interesează rezultatul pentru 1 număr (4) Şirul are 2 termeni: 1 şi 1 $(1 + 1)(1 + 1) = 2 * 2 = 4$

Timp maxim de execuție: 0.1 secunde / test
Memorie totală disponibilă: 2MB şi 1MB pentru stivă
Dimensiunea maximă a sursei: 5KB