

1.Feladat – Csomagok

100 pont

A VII. és VIII. osztály tanulói egy jótékonyági akción fognak részt venni, amelyen cukorka-csomagokat szeretnének ajándékozni a település árvaházának. Ehhez az akcióhoz ők n doboz, különböző típusú cukorkát vásároltak; minden dobozban b_i darab cukorka található.

Kérelmek

Írjunk egy programot, a mely meghatározza a következőket:

- maximum hány cukorka-csomagot lehet összeállítani úgy, hogy mindegyik egyforma legyen, vagyis minden csomag egy adott cukorka típusból azonos számú cukorkát tartalmazzon, minden csomagban legyen minden cukorka-típusból és egyetlen cukorka se maradjon meg.(mindegyik bekerüljön csomagba);
- össze lehet-e állítani n egyforma cukorka-csomagot úgy, hogy minden csomagban azonos számú cukorka legyen minden típusból és legalább egy doboz összes cukorkáját felhasználjuk. Határozzuk meg, hány cukorka nem kerül be így csomagba.

Bemeneti adatok:

A *pachete.in* állományból kiolvassuk a következőket:

- az első sorból egy n természetes számot, amely a cukorka-dobozok számát jelzi
- a következő sorból n természetes számot, amelyek egymás között szóközzel vannak elválasztva. Ezek a számok egy-egy dobozban lévő cukorkák darabszámát jelentik.

Kimeneti adatok:

A *pachete.out* állományba írjuk a következőket:

- az első sorába egy természetes számot, amely megadja, hogy maximum hány egyforma csomagot készíthetünk úgy, hogy minden csomag egy adott cukorka típusból azonos számú cukorkát tartalmazzon, minden csomagban legyen minden cukorka-típusból és egyetlen cukorka se maradjon meg (mindegyik bekerüljön csomagba);
- a második sorba a következőket írjuk:
 - a DA szót, egy szóközt és egy természetes számot, a szám a dobozokban maradt, csomagokba be nem került cukorkák számát jelenti
 - vagy a NU szót, egy szóközt, és a dobozokban található cukorkák összegét írjuk, mivel egyetlen cukorka sem kerül be csomagba, így mind megmarad.

Megkötések

- $1 < n \leq 1000$
- $1 \leq b_i \leq 10000$
- Az a) pontnak megfelelő kérelmek helyes megoldása esetén a pontszám 50% -a.
- A b) pontnak megfelelő kérelmek helyes megoldása esetén a pontszám 50% -a.

Példa

<i>pachete.in</i>	<i>pachete.out</i>	Magyarázat
4 24 18 12 30	6 DA 36	4 doboz cukorka van Az 1. doboz 24 cukorkát, a 2. doboz 18 cukorkát, a 3. doboz 12 cukorkát, míg a 4. doboz 30 cukorkát tartalmaz. 6 egyforma csomag állítható össze. DA, mivel összeállítható 4 darab olyan csomag, amely minden cukorka típusból azonos számú cukorkát tartalmaz, és legalább egy doboz cukorkát teljesen felhasználunk hozzá. 36 darab cukorka nem fog bekerülni csomagokba.
5 24 18 12 30 12	6 NU 96	5 doboz cukorka van 6 egyforma csomag állítható össze. NU mivel nem állítható össze 5 csomag, amely minden cukorka típusból azonos számú cukorkát tartalmaz, és legalább egy doboz cukorkát teljesen felhasználunk hozzá, vagyis mind a 96 darab cukorka megmarad.

Maximális lefutási idő/teszt: 0.6 másodperc

Rendelkezésre álló memória 2 MB amelyből 1 MB a veremnek.

A forrásállomány maximális mérete 5 KB.

2. Feladat – fénykép

100 pont

Egy fekete-fehér képet kódolva mentünk le egy kétdimenziós négyzetes tömbbe, amelynek mérete n . A tömb elemei 0 és 50 közötti számok. A fehér színt a 0 értékkel, a feketét 50-es értékkel kódoltuk, míg a szürke szín különböző árnyalataihoz az 50-es értéktől kisebb 0-tól különböző természetes számokat használunk. A képre a következő átalakítások alkalmazhatók:

1. típusú átalakítás: FORGATÁS – olyan művelet, amelynek eredménye a kép 90° -al való elfordítása az óramutató járásával megegyező irányban

2. típusú átalakítás: ZOOM – olyan művelet, amelynek eredménye a megadott kép $2 \times n$ méretre való felnagyítása úgy,

hogy az x kétdimenziós tömb minden elemét átalakítjuk $\begin{pmatrix} X & X \\ X & X \end{pmatrix}$ típusúra. Például egy olyan fénykép esetén, amely

a $\begin{pmatrix} 1 & 0 \\ 2 & 5 \end{pmatrix}$ kétdimenziós tömbbel van lekódolva, ha alkalmazzuk az 1. típusú átalakítást, vagyis a FORGATÁS-t, a

következőt kapjuk: $\begin{pmatrix} 2 & 1 \\ 5 & 0 \end{pmatrix}$.

Ha az $\begin{pmatrix} 1 & 0 \\ 2 & 5 \end{pmatrix}$ tömbre 2. típusú átalakítást alkalmazunk, vagyis ZOOM -ot, a következőt kapjuk: $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 2 & 5 & 5 \\ 2 & 2 & 5 & 5 \end{pmatrix}$

Kérelmek

Ha meg van adva, egy "1" és "2" átalakításokból álló művelet sor, amelyet egymás után kell végrehajtani balról jobbra haladva, írjuk ki a végső kép méretét és kódolását, az össze kért átalakítás végrehajtása után.

Bemeneti adatok

A **poza.in** bemeneti állományból a következőket olvassuk ki:

- az első sorából egy n természetes számot, amely a megadott kép méretét jelöli;
- a következő n sor mindegyikéből n különböző, egymástól szóközzel elválasztott értéket. Ezek az értékek a kép kódolásához felhasznált kétdimenziós tömb elemeinek felelnek meg;
- a következő sorból egy k természetes számot, amely megadja hány darab átalakítást kell majd végrehajtani a képen;
- az állomány következő sorából, k darab 1 és 2 –ből álló, egymástól szóközzel elválasztott számokat, amelyek a képre alkalmazandó átalakításokat jelentik. Az átalakításokat abban a sorrendben kell végrehajtani ahogyan meg vannak adva, balról jobbra haladva.

Kimeneti adatok

A **poza.out** állomány első sorába egyetlen természetes számot kell beírni, annak a képnek a méretét, amelyet az átalakítás sorozat alkalmazása után kapunk.

A következő n sor mindegyikébe n darab egymástól szóközzel elválasztott értéket írunk, a végső kép kódolását.

Megkötések és pontosítások

- $0 < n \leq 150$, $1 \leq k \leq 99$
- a végső kép mérete az átalakítások alkalmazása után mindig kisebb lesz mint 150

Példa

poza.in	poza.out	Magyarázat
2	4	A kép mérete 2 és 3 átalakítást alkalmazunk rá: 1 1 2, vagyis FORGATÁS, FORGATÁS és ZOOM
1 0	5 5 2 2	-ot.
2 5	5 5 2 2	Az első FORGATÁS után a: $\begin{pmatrix} 2 & 1 \end{pmatrix}$ képet, majd a második FORGATÁS után a $\begin{pmatrix} 5 & 2 \end{pmatrix}$ képet
3	0 0 1 1	kapjuk.
1 1 2	0 0 1 1	$\begin{pmatrix} 5 & 0 & & \\ & & & 0 & 1 \end{pmatrix}$
		Ezt követően alkalmazzuk a ZOOM átalakítást és megkapjuk a végső kép kódolását.
		5 5 2 2

		5 5 2 2
		0 0 1 1
		0 0 1 1

Maximális lefutási idő/teszt: 1 másodperc
Összmemória: 2 MB amelyből 1 MB a veremnek.
A forrásállomány maximális mérete 5 KB.

Descrierea problemei pachete

Pentru a determina cel mai mare număr de pachete egale care se pot realiza utilizând același număr de bomboane din același sortiment și toate sortimentele, trebuie să determinăm cel mai mare divizor comun al tuturor celor n numere. Acesta reprezintă numărul maxim de pachete care se pot forma cu toate bomboanele. Pentru a realiza n pachete cu bomboane, fiecare având același număr de bomboane din fiecare sortiment, epuizând cel puțin o cutie, vom căuta cutia care are un număr **minim** de bomboane.

Dacă acest număr este divizibil cu n , atunci se pot forma n pachete. Pentru a calcula bomboanele care rămân trebuie să calculăm suma:

$$\sum_{i=0}^{n-1} (b[i] - \text{minim})$$

Dacă **minim** nu se divide la n înseamnă că nu vom putea realiza n cutii egale cu număr egal de bomboane din fiecare sortiment și să epuizăm bomboanele din cel puțin o cutie.

Descriere soluție Problema 2 -poza

*Prof. Iordaiache Cristina
Lic. Teoretic „Grigore Moisil” Timișoara*

- Se citesc datele din fișierul de intrare memorandu-se într-un tablou bidimensional de dimensiune n codificarea inițială a pozei
- Se citesc apoi cele n transformări aplicate pozei
- Pentru fiecare transformare de tip 1-ROTIRE se rotește spre dreapta cu 90 de grade tabloul bidimensional
- Pentru fiecare transformare de tip 2-ZOOM se va opera în tabloul curent modificarea corespunzătoare fiecărui element din tablou astfel încât,

fiecare element X al tabloului se va transforma în $\begin{pmatrix} X & X \\ X & X \end{pmatrix}$

```
for(i=1;i<=n;i++)
  for(j=1;j<=n;j++)
  {
    b[2*i-1][2*j-1]=a[i][j];
    b[2*i-1][2*j]=a[i][j];
    b[2*i][2*j-1]=a[i][j];
    b[2*i][2*j]=a[i][j];
  }
```