

Lepota algoritama deo 1

1.

Dat je niz A ($1 \leq A[i] \leq 1.000$) prirodnih brojeva duzine N ($1 \leq N \leq 1.000.000$). Naci broj podnizova (uzastopni elementi) cija je suma parna.

Prvi red standardnog ulaza sadrzi prirodni broj N koji predstavlja broj elemenata niza. Naredni red sadrzi N prirodnih brojeva, razdvojenih jednim znakom razmaka, koji predstavljaju elemente niza.

Prvi i jedini red standardnog izlaza treba da sadrzi jedan prirodan broj - broj podnizova parne sume.

Primer 1

Ulaz	Izlaz
4 1 2 3 4	4

Ograničenja

Vremensko ograničenje: 1 sekunda

Memorijsko ograničenje: 64 MB

Rešenje 1 (neefikasan algoritam s obzirom da $N \leq 1.000.000$)

Za svaku vrednost promenljivih i, j , sumirati elemente sa indeksima između njih i proveriti da li je to paran broj.

Posmatrajmo pseudokod za ovu ideju prikazan u Algoritmu:

```
=====
brojParnih = 0;
for i = 1 to n
  for j = i to n
    suma = 0;
    for k = i to j do
      suma = suma + a [k];
      if (suma mod 2 == 0) then brojParnih = brojParnih + 1;
```

Složenost Algoritma je $O(n^3)$

Rešenje 2:

Pokušajmo prethodnu implementaciju da ubrzamo. Linije koda koje sumiraju vrednosti elemenata od i do j mogu se drukčije implementirati.

Kada fiksiramo levu granicu tog segmenta po kome sumiramo, promenjivu i , posmatrajmo kako se menja suma kada se promenjiva j povećava za 1.

Primećujemo da se suma poveća za vrednost $a[j]$.

Dakle, NE moramo uvek sumirati sve elemente počev od indeksa i . Implementacija ovde ideje je prikazana u Algoritmu koji sledi.

```
brojParnih = 0;
for i = 1 to n
  suma = 0;
  for j = i to n
    suma = suma + a [j];
    if (suma mod 2 == 0) then brojParnih = brojParnih + 1;
```

Složenost novog algoritma je $O(n^2)$.

Međutim, ovaj problem možemo rešiti i u linearnom vremenu. Definišimo niz *suma* kao

$$suma[k] = a[1] + \dots + a[k]$$

Sumu elemenata $a[i] + \dots + a[j]$ možemo preko niza *suma* izračunati kao $suma[j] - suma[i - 1]$, gde uzimamo da je $suma[0] = 0$.¹ Ovaj podniz će imati parnu sumu jedino u slučaju da su vrednosti $suma[j]$ i $suma[i - 1]$ iste parnosti. Označimo sa *m* broj parnih elemenata niza *suma*, računajući i element sa indeksom 0. Ukoliko posmatramo dva parna elementa, podniz koji oni definišu (za koji su oni leva i desna granica) je parne sume. Analogno imamo i sa neparnim elementima. Kako je razlika dva broja parna ako i samo ako su članovi iste parnosti, ovo su jedini mogući slučajevi parnih podnizova. Krajnji rezultat dobijamo kao

$$\binom{m}{2} + \binom{n+1-m}{2}$$

Složenost algoritma je linearna. Niz *suma* možemo da inicijalizujemo jednim prolaskom kroz niz vezom:

$$suma[0] = 0$$

$$suma[k] = suma[k - 1] + a[k], \quad k \geq 1$$

Nakon toga jednostavnim prebrojavanjem parnih odnosno neparnih elemenata niza, rešenje dobijamo gornjoj formulom.

Rešenje:

```
#include <stdio.h>
#define MAX 1000000

int n, a[MAX], thisSum, brNeparnih, brParnih;

int main() {
    int i;
    long long rez;
    scanf("%d", &n);
    scanf("%d", &a[0]);
    thisSum = a[0];
    if (a[0]%2) brNeparnih++;

    for (i = 1; i < n; i++) {
        scanf("%d", &a[i]);
        thisSum += a[i];
        if (thisSum%2) brNeparnih++;
    }

    brParnih = n - brNeparnih + 1;
    rez = (long long)brParnih*(brParnih-1)/2 + (long long)brNeparnih*(brNeparnih-1)/2;

    printf("%lld", rez);
```

```
return 0;
```

```
}
```

2. (1 poen)

Svake godine, Beograd je domaćin festivala horova - popularan festival koji privlači horove i pevačke formacije iz različitih zemalja. Organizatori festivala horova 2013 su izabrali nekoliko lepih devojaka da budu vodiči tokom festivala. Vodiči su zaduženi da pomažu učesnicima tokom boravka u Beogradu.

Kao što znate, dobra je praksa da svi učesnici iz neke zemlje imaju jednog vodiča- kako bi zajedno šetali i obedovali. Međutim, Marko, jedan od organizatora festivala, nije bio tako mudar – napravio je veliku zbrku i svakom učesniku festivala horova je saopštio nasumično ime vodiča. Tako da, Vaš posao je da pomognete Marku da odredi minimalni broj učesnika koje mora da obavesti (da imaju novog vodiča) kako bi se popravio problem. Pri tome važi: da svaki voduč bude zadužen za učesnike iz najviše jedne zemlje, kao i da svi učesnici iz iste zemlje imaju istog vodiča. Ako postoji više vodiča nego zemalja učesnica, onda neki vodiči mogu ostati i bez zaduženje.

Ulaz

Prva linija ulaza sadrži dva cela broja N ($1 \leq N \leq 100$) i M ($1 \leq M \leq 15$ and $M \leq N$), gde N je broj vodiča i M je broj zemalja učesnica festivala horova 2013.

Narednih N linija opisuje broj učesnika za koje je zadužen svaki vodič: svaka linija sadrži M celih brojeva X_{ij} ($0 \leq X_{ij} \leq 10$), gde X_{ij} je broj učesnika iz j -te zemlje za koje je zadužen i -ti vodič (prvi red sadrži $X_{11}, X_{12}, X_{13}, \dots, X_{1M}$, drugi red sadrži $X_{21}, X_{22}, X_{23}, \dots, X_{2M}$, etc). Postoji bar 2 učesnika iz svake zemlje na tekućem festivalu horova.

Izlaz

Vaš program mora da ispiše tačno jedan ceo broj - zahtevani minimalni broj učesnika koje Marko mora da obavesti.

Ograničenja

Vremensko ograničenje: 1 sekunda

Memorijsko ograničenje: 64 MB

Primer 1

input	Output
3 2	5
4 9	
1 0	
2 0	

Objašnjenje za primer 1: Marko treba da obavesti 5 učesnika. Nakon reorganizacije, vodič broj 1 će biti odgovoran za 9 učesnika (svi iz zemlje broj 2), a vodič broj 3 će biti odgovoran za $4+1+2=7$ učesnika (svi iz zemlje broj 1).