

Srpska informatička olimpijada, Beograd, 16. maj 2015.

U svakom zadatku važe ograničenja: Vremensko ograničenje 1s

Memorijsko ograničenje 64 MB

1. Ana i Boris igraju igru slaganja pez bombonica. Na početku igre data je kula koju čini N bombonica i na početku Ana i Boris biraju dva različita pozitivna cela broja P i T . Igru započinje Ana, potom igra Boris i tako redom. Osoba koja je na redu da igra može sa kule uzeti 1, P ili T bombonica. Pobjednik je osoba koja uzme poslednju bombonicu (ili bombonice). Nakon mnogobrojnih odigranih partija, Ana je uočila da postoje situacije kada ona može biti pobjednik, bez obzira na potez koji odigra Boris. Takođe važi i obratno, postoje situacije kada Boris može da pobjedi bez obzira na to kako će Ana odigrati. Stoga pre početka igre, Ana je nestrpljiva da sazna ishod igre. Napišite program **SLAGANJE** koji će pomoći Ani da predvidi rezultat igre za date brojeve P , T i N .

ULAZ

Na ulazu je dat opis m igri. U prvoj liniji standardnog ulaza dati su celi brojevi P , T i m , $1 < P < T < 10$, $3 < m < 50$. U drugoj liniji standardnog ulaza dato je m celih brojeva N_1, N_2, \dots, N_m , $1 \leq N_i \leq 1000000$, $i = 1, 2, \dots, m$, koji predstavljaju broj bombonica u svakoj od m kula.

IZLAZ

Na standardnom izlazu ispisati nisku (string) koji se sastoji od m slova A i B . Ako je Ana pobjednik u i -toj igri (bez obzira na to kako igra Boris), onda i -to slovo niske mora biti A (veliko slovo A). Ako je Boris pobjednik u i -toj igri (bez obzira na to kako igra Ana), onda i -to slovo u nisci mora biti B .

PRIMER

Ulaz

2 3 5

3 12 113 25714 88888

Izlaz

ABAAB

2. Definišimo rastojanje između dva prirodna broja kao sumu apsolutnih vrednosti razlika njihovih cifara (rastojanje brojeva 8752 i 1589 se računa kao $|8-1|+|7-5|+|5-8|+|2-9|=19$). Ako jedan od brojeva ima manje cifara od drugog, smatramo da su tom broju dopisane vodeće nule (rastojanje brojeva 32 i 1234 se računa kao $|0-1|+|0-2|+|3-3|+|2-4|=5$). Dati su prirodni brojevi A i B . Napišite program **RASTOJANJE** koji će izračunati sumu udaljenosti svih parova brojeva iz intervala $[A, B]$.

ULAZ

U jedinju liniji standardnog ulaza zadati su prirodni brojevi A i B razdvojeni blanko karakterom ($1 \leq A \leq B \leq 10^{50000}$).

IZLAZ

U jedinu liniju standardnog izlaza upisati jedan ceo broj – izračunati broj parova. Taj broj može biti veoma velik, te ispišite njegov ostatak pri deljenju sa 1 000 000 007.

PRIMER

Ulaz

178 181

Izlaz

76

Objašnjenje: Rastojanja su redom $(178, 179) = 1$, $(178, 180) = 9$, $(178, 181) = 8$, $(179, 180) = 10$, $(189, 191) = 9$, $(180, 181) = 1$. Svako od rastojanja se broji dvaput, što ukupno daje $2 * (1 + 9 + 8 + 10 + 9 + 1) = 76$

3. Dat je skup od n različitih pozitivnih celih brojeva a_1, a_2, \dots, a_n , čije su vrednosti između 1 i 1000000 i ceo broj s . Napisati program **SUMACLAN** koji izračunava broj parova (a_i, a_j) , gde $1 \leq i < j \leq n$ i važi da $a_i + a_j = s$.

ULAZ

U prvoj liniji standardnog ulaza nalazi se ceo broj n ($1 \leq n \leq 100000$). U drugoj liniji standardnog ulaza nalazi se n celih brojeva, članovi skupa, razdvojenih blanko karakterom. U trećoj liniji standardnog ulaza dat je ceo broj s ($1 \leq s \leq 2000000$).

IZLAZ

Napišite broj parova na standardni izlaz.

PRIMER

Ulaz

9

9 12 5 10 7 1 11 3 2

13

Izlaz

3

Objašnjenje: Postoje tri para čija suma je jednaka 13: $(12, 1)$, $(10, 3)$ i $(2, 11)$.

1. SLAGANJE

```
#include <stdio.h>
#define Pmax 1000001
#define Umax 11
char pobeda[Pmax];
int ucitaj[Umax];
int main()
{ int P,T,M,Nmax=0,i;
  int pobednicka,pe,te;

scanf("%d %d %d",&P,&T,&M);

// ucitavanje igara i pronalazenje maksimalnog N, tj. Nmax
for(i=1;i<=M;i++)
{ scanf("%d",&ucitaj[i]);
  if(Nmax<ucitaj[i]) Nmax=ucitaj[i];
}

pobeda[0]=0; // pozicija 0 je gubitnicka
for(i=1;i<=Nmax;i++)
{
pobednicka=pobeda[i-1];
if(i-P>=0) pe=pobeda[i-P]; else pe=1;
if(i-T>=0) te=pobeda[i-T]; else te=1;
if(pobednicka&&pe&&te) pobeda[i]=0; else pobeda[i]=1;
}

for(i=1;i<=M;i++)
if(pobeda[ucitaj[i]]) printf("A"); else printf("B");
printf("\n");
return 0;
}
```

2. RASTOJANJE

```
#include <cstdio>
#include <cstring>
#include <algorithm>

using namespace std;

#define Ncif 50010
#define OST 1000000007;

int N;
char A[Ncif], B[Ncif];
int mem1[2][2][2][2][Ncif];
int mem2[2][2][2][2][Ncif];

void ucitajSaNulama(void) {
  char pom1[Ncif], pom2[Ncif];
  scanf("%s%s", pom1, pom2);

  // ako je potrebno, dopunjavanje nize donje granice vodecim nulama
```

```

int duz1 = strlen(pom1), duz2 = strlen(pom2);
for(int i = 0; i < duz2 - duz1; ++i) A[i] = '0';

for(int i = duz2 - duz1; i < duz2; ++i)
    A[i] = pom1[i - duz2 + duz1];

for(int i = 0; i < duz2; ++i)
    B[i] = pom2[i];

N = duz2;
}

int krajeviBroja(int pos, int d1, int g1, int d2, int g2) {
    // d1 -> indikator da li je prvi broj dosao do donje granice
    // g1 -> indikator da li je prvi broj dosao do gornje granicu
    // d2, g2 -> indikator za drugi broj na gore opisani nacin

    if(pos == N) return 1;

    int &dinstanje = mem1[d1][g1][d2][g2][pos];
    if(dinstanje != -1) return dinstanje;

    int lo1 = 0, hi1 = 9, lo2 = 0, hi2 = 9;
    // ako je d1 = g1 = 0, onda trenutna cifra moze biti sve od 0 do 9

    if(d1 == 1) {
        // prvi broj trenutno dosegao donju granicu, te mu
        // trenutna cifra ne sme biti manja od iste
        // cifre u donjoj granici
        lo1 = A[pos] - '0';
    }

    if(g1 == 1) {
        // prvi broj trenutno dosegao gornju granicu, te mu
        // trenutna cifra ne sme biti veca od iste
        // cifre u gornjoj granici
        hi1 = B[pos] - '0';
    }

    // analogno za drugi broj
    if(d2 == 1) lo2 = A[pos] - '0';
    if(g2 == 1) hi2 = B[pos] - '0';

    long long rez = 0;

    for(int x = lo1; x <= hi1; ++x) {
        for(int y = lo2; y <= hi2; ++y) {
            int nd1 = 0, ng1 = 0, nd2 = 0, ng2 = 0;
            // proveriti granice
            if(d1 == 1 && x == lo1) nd1 = 1;
            if(g1 == 1 && x == hi1) ng1 = 1;
            if(d2 == 1 && y == lo2) nd2 = 1;
            if(g2 == 1 && y == hi2) ng2 = 1;
            rez += krajeviBroja(pos + 1, nd1, ng1, nd2, ng2);
        }
    }
}

```

```

    }
}

return dinstanje = rez % OST;
}

int f(int pos, int d1, int g1, int d2, int g2) {
    // d1, g1, d2, g2 -> isto kao u gornjoj funkciji

    if(pos == N) return 0;
    int &dinstanje = mem2[d1][g1][d2][g2][pos];
    if(dinstanje != -1) return dinstanje;

    int lo1 = 0, hi1 = 9, lo2 = 0, hi2 = 9;

    // sve isto kao i gore
    if(d1 == 1) lo1 = A[pos] - '0';
    if(g1 == 1) hi1 = B[pos] - '0';
    if(d2 == 1) lo2 = A[pos] - '0';
    if(g2 == 1) hi2 = B[pos] - '0';

    long long rez = 0;

    for(int x = lo1; x <= hi1; ++x) {
        for(int y = lo2; y <= hi2; ++y) {
            // sve isto kao i gore
            int nd1 = 0, ng1 = 0, nd2 = 0, ng2 = 0;
            if(d1 == 1 && x == lo1) nd1 = 1;
            if(g1 == 1 && x == hi1) ng1 = 1;
            if(d2 == 1 && y == lo2) nd2 = 1;
            if(g2 == 1 && y == hi2) ng2 = 1;
            rez += (long long)abs(x - y) * krajeviBroja(pos + 1, nd1, ng1, nd2, ng2) % OST;
            rez += f(pos + 1, nd1, ng1, nd2, ng2);
        }
    }

    return dinstanje = rez % OST;
}

int main(void) {
    ucitajSaNulama();
    memset(mem1, -1, sizeof mem1);
    memset(mem2, -1, sizeof mem2);
    printf("%d\n", f(0, 1, 1, 1, 1));
    return 0;
}

```

3. SUMACLAN

```

#include <iostream>
#include <cstdio>
#include <vector>
#include <algorithm>

```

```

using namespace std;

```

```
int main ()
{
int n, b, s, br=0;
vector <int> a;
scanf ("%d", &n);

for (int i = 0; i < n; i++){
scanf ("%d", &b);
a.push_back (b);
}

scanf ("%d", &s);

sort(a.begin(),a.end());

int k=0, m=n-1;
while (k<m){
if (a[k]+a[m]==s){br++; k++; m--;}
else if (a[k]+a[m]<s)k++;
else if (a[k]+a[m]>s)m--;
}

printf("%d\n", br);
return 0;
}
```