

Խնդիր 1. Մեծագույն եռանկյուն

Տրված են հատվածներ: Հարկավոր է ընտրել երեք այնպիսի հատվածներ, որոնցից հնարավոր լինի կազմել մեծագույն հնարավոր մակերեսով եռանկյուն:

Մուտքը

Առաջին տողում տրված է եռանկյունների n քանակը ($n \leq 20$): Երկրորդ տողում տրված են 1000-ին չգերազանցող n բնական թվեր՝ հատվածների երկարությունները:

Ելքը

Ելքային ֆայլի առաջին տողում հարկավոր է արտածել մեծագույն եռանկյան մակերեսը, որ կարելի է կառուցել տրված հատվածներից: Եթե հնարավոր չէ կառուցել ոչ զրոյական մակերեսով եռանկյուն, արտածել -1 թիվը:

Օրինակ

Մուտքը	Ելքը
3	0.4330127
1 1 1	

Լուծումը

Սա մեծագույն տարրը գտնելու խնդիր է: Հարկավոր է դիտարկել բոլոր եռյակները և գտնել մեծագույն մակերեսը Հորների բանաձևով: Միայն թե պետք է նախքան մակերեսը հաշվելը ստուգել տրված երեք կողմերով եռանկյուն կառուցելու հնարավորությունը: Ներկայացնում ենք “Քվանտ” վարժարանի 7-րդ դասարանի աշակերտ Գրիգորյան Էդուարդի ծրագիրը.

```
VAR N,a,b,c,i,i1,i2,h1,z,z1,z2:longint;  
s,p,rec:real;  
m:array [1..20] of integer;  
BEGIN  
  readln(N);  
  rec:=0;  
  h1:=0;  
  for i:=1 to N do read(m[i]);  
  for i:=1 to N do  
    for i1:=i+1 to N-1 do  
      for i2:=i1+1 to N do begin  
        a:=m[i];  
        b:=m[i1];  
        c:=m[i2];  
        if (a+b>c) and (a+c>b) and (b+c>a) then begin  
          p:=(a+b+c)/2;  
          h1:=1;  
          s:=sqrt(P*(p-a)*(P-b)*(P-c));  
          if s>rec then rec:=s;  
        end;  
      end;  
  if h1=0 then rec:=-1;  
  writeln(rec:1:7);  
end.
```

Խնդիր 2. Հավասարեցում

Տրված է մետաղադրամներով N պարկ: Առաջին պարկում կա մեկ մետաղադրամ: Երկրորդ պարկում կա երկու մետաղադրամ, և այլն, K համարի պարկում կա K մետաղադրամ: Ամեն քայլին թույլատրվում է ցանկացած L թվով պարկերից յուրաքանչյուրից հանել S մետաղադրամ: (Յուրաքանչյուր քայլին L -ը և S -ը կարող են փոխվել):

Գրել ծրագիր, պարզելու համար, թե առնվազն քանի քայլ է պետք կատարել պարկերում մետաղադրամների քանակները հավասարեցնելու համար:

Մուտք

Մուտքում տրված է մի ամբողջ թիվ՝ պարկերի N ($0 < N < 10^9$) քանակը:

Ելք

Ելքում պետք է արտածել մի թիվ՝ մինիմալ քայլերի քանակը, որոնց միջոցով կարելի է մետաղադրամներ հանել այնպես, որ արդյունքում յուրաքանչյուր պարկում մնա ճիշտ մեկ մետաղադրամ:

Օրինակ

Մուտք	Ելք
3	2

Լուծումը

Սկզբում բոլոր պարկերում տարբեր թվով մետաղադրամներ կան: Եթե որևէ քայլից առաջ ունենք հավասար քանակով մետաղադրամներ պարունակող մեկից ավել թվով պարկեր, ապա նրանցից կարելի է թողնել միայն մեկը. Եթե այդ պարկից ինչ-որ քանակությամբ մետաղադրամներ հանենք, նույնքան էլ կհանենք իրեն հավասար մյուս պարկերից: Այդ դեպքում կարելի է ցույց տալ հետևյալը. եթե ունենք N տարբեր պարկեր, ապա հաջորդ քայլին նրանց քանակը կլինի առնվազն $\lceil N/2 \rceil$ (այստեղ $\lceil \cdot \rceil$ նշանները ցույց են տալիս, որ պետք է կլորացնել դեպի վեր): Իրոք, եթե L պարկերից մետաղադրամներ են հանվում, ապա տարբեր քանակով մետաղադրամներ պարունակող պարկերի թիվը կլինի առնվազն $\max(N-L, L)$: Այսպիսով, Առաջին քայլից հետո կարելի է ստանալ $\lceil N/2 \rceil$ տարբեր պարկեր, մետաղադրամներ հանելով $N/2$ -ից մեծ համար ունեցող բոլոր պարկերից: Կարելի է ցիկլ կազմակերպել և N -ը շարաունակաբար բաժանելով հաշվել քայլերի քանակը, կամ կարելի է, ի վերջո, գալ մի բանաձև .

$\lceil \log N \rceil$:

Ներկայացնում ենք “Քվանտ” վարժարանի 11-րդ դասարանի աշակերտ Չատրյան Ալեքսանդրի ծրագիրը.

```
#include <iostream>
#include <cmath>
using namespace std;

int main ()
{
    freopen ("input.txt", "r", stdin);
    freopen ("output.txt", "w", stdout);

    int n=0;
    scanf ("%d", &n);
    int s=0, m=n;
    while (m!=1)
    {
```

```
        m=(m+1)/2;  
        s++;  
    }  
    printf ("%d\n",s);  
    return 0;  
}
```

Խնդիր 3. Անունների ձուլում

Տրված է երկու անուն: Հարկավոր է գտնել մինիմալ երկարության այնպիսի տող, որի համար երկու անուններն էլ հանդիսանան ենթատող, այսինքն տողը սկսվում է անուններից մեկով և ավարտվում է մյուսով:

Մուտքը

Մուտքում տրված է երկու տող, որոնցից յուրաքանչյուրը պարունակում է մի անուն: Անունները բաղկացած են անգլերեն տառերից, որոնցից միայն առաջինն է մեծատառ: Անունների երկարությունները չեն գերազանցում 30-ը:

Ելքը

Ելքում հարկավոր է արտածել ամենակարճ տողը, որում միաժամանակ հանդիպում են մուտքում տրված անունները: Այդ տողում անունները պետք է սկսվեն մեծատառով: Եթե գոյություն ունի մի քանի լուծում, հարկավոր է արտածել այն, որը բառարանային կարգով ավելի փոքր է:

Օրինակ

Մուտքը	Ելքը
Arman Mane	ArMane

Լուծումը

Այս խնդրի ալգորիթմը բարդ չէ, դժվարությունը այն խնամքով իրականացնելու մեջ է: Պետք է հատարկելով գտնել երկրորդ բառի լավագույն տեղադրումը առաջին բառի վերջում, ապա առաջին բառի լավագույն տեղադրումը երկրորդ բառի վերջում և ընտրել դրանցից կարճը, իսկ հավասար երկարության դեպքում՝ նրանցից բառարանային կարգով առաջինը: «Քվանտ» վարժարանի 11-րդ դասարանի աշակերտ Տեփանյան Հայկի ծրագիրը.

```
#include<iostream>
#include<algorithm>
#include<cstring>
using namespace std;

int sa1,sa2,p1,p2,p0;
char a1[82],a2[82],c1[82],c2[82];
bool f;
int main()
{
    freopen("input.txt","r",stdin);
    freopen("output.txt","w",stdout);
    int i;
    cin>>a1>>a2;
    sa1=strlen(a1);
    sa2=strlen(a2);
    p1=sa1;
    for(p0=sa1-1;p0>=0 && sa1-p0<=sa2;p0--)
    {
        f=0;
        for(i=p0;i<sa1;i++)
            if(a1[i]!=a2[i-p0] && abs(a1[i]-a2[i-p0])!=32)
                { f=1; break;}
        if(!f)
            p1=p0;
    }
}
```

```
    }
    for(i=0;i<p1;i++)
        c1[i]=a1[i];
    for(i=p1;i-p1<sa2;i++)
        c1[i]=a2[i-p1];
    c1[sa2+p1]='\0';

    p2=sa2;
    for(p0=sa2-1;p0>=0 && sa2-p0<=sa1;p0--)
    {
        f=0;
        for(i=p0;i<sa2;i++)
            if(a2[i]!=a1[i-p0] && abs(a2[i]-a1[i-p0])!=32)
                { f=1; break;}
        if(!f)
            p2=p0;
    }
    for(i=0;i<p2;i++)
        c2[i]=a2[i];
    for(i=p2;i-p2<sa1;i++)
        c2[i]=a1[i-p2];
    c2[sa1+p2]='\0';

    if(sa2+p1<sa1+p2)
        cout<<c1<<endl;
    if(sa2+p1>sa1+p2)
        cout<<c2<<endl;
    if(sa2+p1==sa1+p2)
    {
        if(strcmp(c1,c2)<=0)
            cout<<c1<<endl;
        else
            cout<<c2<<endl;
    }
    return 0;
}
```

Խնդիր 4. Երկպարզ թվեր

Թիվն անվանենք երկպարզ, եթե այն հանդիսանում է որևէ երկու պարզ թվերի գումար: Պահանջվում է գրել ծրագիր, որը տրված a և b թվերի համար արտածի $[a..b]$ միջակայքին պատկանող երկպարզ թվերի քանակը:

Մուտքը

Մուտքային ֆայլում տրված են երկու ամբողջ a և b ($2 \leq a \leq b \leq 40000$) թվեր:

Ելքը

Ելքային ֆայլում պետք է արտածել աճման կարգով տրված տիրույթին պատկանող երկպարզ թվերի քանակը:

Օրինակ

Մուտքը	Ելքը
3 13	9

Լուծումը

Մրցույթի որոշ մասնակիցներ փորձել են խնդիրը լուծել լրիվ հասարկման եղանակով: Այսինքն, դիտարկել են a -ից b սահմաններում ընկած բոլոր թվերը, յուրաքանչյուր թիվ ներկայացրել են բոլոր հնարավոր եղանակներով երկու թվերի գումարի տեսքով, և եթե այդ թվերը երկուսն էլ պարզ են, երկպարզ թվերի քանակը մեծացրել են մեկով: Արդյունքում ստացվում է դանդաղ ծրագիր: Ավելի արդյունավետ է հետևյալ ալգորիթմը. “Էրատոսթենի մաղ” ալգորիթմով գտնենք և մի զանգվածում պահենք b -ից փոքր բոլոր պարզ թվերը: Մեկ այլ զանգվածում ստանանք այդ պարզ թվերի զույգ առ զույգ գումարները այնպես, որ կրկնվող թվեր չլինեն: Հաշվենք այդ թվերից $[a, b]$ տիրույթին պատկանողների քանակը: Ներկայացնում ենք Երևանի Ա. Շահինյանի անվան ֆիզմաթ դպրոցի 10-րդ դասարանի աշակերտ Վարդանյան Արմենի ծրագիրը.

```
#include <iostream>
#include <fstream>
using namespace std;

const int N=40001;
int a,b,p[N],d[N],c[100000];

int main()
{
    int i, j, q;
    ifstream fin ("input.txt");
    ofstream fout ("output.txt");
    fin>>a>>b;
    q=0;
    for(i=2;i<=b;i++)
    {
        if(p[i]==0)
        {
            for(j=2*i;j<=b;j+=i)
            {
                p[j]=1;
            }
            q++;
        }
    }
}
```

```
        d[q]=i;
    }
}
for(i=1;i<=q;i++)
{
    for(j=i;j<=q;j++)
    {
        c[d[i]+d[j]]=1;
    }
}
q=0;
for(i=a;i<=b;i++)
{
    if(c[i]==1)
    {
        q++;
    }
}
fout<<q;
return 0;
}
```