

```

else // x niama desen brat
{
    sp--; lson[st[sp]]=0;
    if(sp==0) break;
}
}
}
printf("%d\n",L);
return 0;
}

```

Задача А3. ЧЕТВОРКИ, автор Стоян Капралов

Да разгледаме следните 4-елементни множества с елементи цели числа от 1 до 8: {1,2,3,4}, {1,3,5,7}, {2,3,5,8}, {2,3,6,7}, {3,4,5,6}, {5,6,7,8}.

Тази съвкупност притежава свойството, че всеки две множества имат най-много два общи елемента. Може да се докаже, че от множество с 8 елемента, могат да бъдат избрани най-много 14 четворки, така, че всеки две от тях да имат най-много два общи елемента.

Намерете съвкупност от колкото се може повече на брой 4-елементни подмножества на множеството {1,2,3,4,5,6,7,8,9,10}, така, че всеки две от тях да имат най-много два общи елемента.

Резултата запишете във файл **A3.txt**. На първия ред запишете броя **m** на подмножествата в намерената съвкупност. На всеки от следващите **m** реда запишете по едно от намерените подмножества – четири различни цели числа от 1 до 10, разделени с интервали.

Ако записаната във файла съвкупност е коректна, ще получите **100m/M** точки, където **M** е броя на четворките в съвкупността, построена от журито.

Решение

Условието две четворки да имат най-много два общи елемента означава, че нямат обща тройка. Всяка четворка „покрива” 4 различни тройки. Следователно в решение с **m** четворки ще бъдат покрити общо **4m** различни тройки. Всички възможни тройки са $\binom{10}{3} = 120$. Следователно $4m \leq 120$,

откъдето $m \leq 30$.

Ако съществува решение с 30 четворки, всяка тройка ще се среща точно веднъж в някоя от четворките. Такава конфигурация се нарича шайнерова система от четворки.

Ще опитаме да построим решение с 30 четворки. Всички четворки са $\binom{10}{4} = 210$. Избирането на 30 от тях, без използването на някакви допълнителни съображения е практически неизпълнимо.

Нека **k** е броят на всички четворки в решението, съдържащи числото 1. Всяка такава четворка покрива по 3 двойки от множеството {2,3,4,5,6,7,8,9,10}, а две такива четворки не бива да имат обща двойка, защото вече имат общ елемент – числото 1. Общо двойките, които могат да се образуват от 9 числа са 36. Следователно $k \leq 12$.

Всички четворки, съдържащи числото 1 са $\binom{9}{3} = 84$. От тях искаме да

изберем 12, така че да са съвместими помежду си. Това вече става за секунди. Едно възможно решение е:

{1,2,3,4} {1,2,5,6} {1,2,7,8} {1,2,9,10} {1,3,5,7} {1,3,6,9}
 {1,3,8,10} {1,4,5,10} {1,4,6,8} {1,4,7,9} {1,5,8,9} {1,6,7,10}.

Остава да продължим решението с още 18 четворки, несъдържащи числото 1. Общият брой на четворките, от които ще избираме е $210 - 84 = 126$.

Оказва се, че с праволинейно написана програма, без някакви допълнителни „хитрости” намираме решение с 30 четворки за не повече от 10 секунди.

```

#include <fstream>
using namespace std;
int a[210][4];
int c[30];
void init()
{
    int p=0;
    for(int i0=1; i0<=7; i0++)
        for(int i1=i0+1; i1<=8; i1++)
            for(int i2=i1+1; i2<=9; i2++)
                for(int i3=i2+1; i3<=10; i3++)
                    { a[p][0]=i0; a[p][1]=i1; a[p][2]=i2; a[p][3]=i3;
                      p++;}
}

bool ok2(int i1, int i2)
{
    int s=0;
    for(int j1=0; j1<4; j1++)
        for(int j2=0; j2<4; j2++)
            if(a[i1][j1]==a[i2][j2]) s++;
    if(s>2) return false;
    return true;
}

bool ok(int k)
{
    for(int i=0; i<k; i++)
        if(!ok2(c[i],c[k])) return false;
    return true;
}

void show()
{
    ofstream fout("A3.txt");
    fout << 30 << endl;
    for(int i=0; i<30; i++)
        fout << a[c[i]][0] << " " << a[c[i]][1] << " "
            << a[c[i]][2] << " " << a[c[i]][3] << endl;
    fout.close();
    exit(0);
}

```

```

void gen(int k)
{ if(k==30) show();
  int min,max;
  if(k==0) min=0; else min=c[k-1];
  if(k<12) max=84; else max=210;
  for(c[k]=min; c[k]<max; c[k]++)
    if(ok(k)) gen(k+1);
}
int main()
{ init();
  gen(0);
}

```

Група В

Задача В1. СУМИ, автор Младен Манев

Напишете програма **SUM**, която да намира броя на начините, по които зададено цяло положително число N може да се представи като сума на поне две последователни положителни цели числа. Числото 5, например, може да се представи като такава сума точно по един начин: $5=2+3$. Числото 8, пък, изобщо не може да се представи така (0 начини), а числото 21 има три такива представяния: $21=10+11=6+7+8=1+2+3+4+5+6$.

Вход

Програмата трябва да въвежда цяло число N ($2 < N < 10^8$) от стандартния вход.

Изход

Програмата трябва да изведе на стандартния изход броя на начините, по които N може да се запише като сума на поне две последователни положителни цели числа.

ПРИМЕР

Вход	Изход
9	2

Решение

Нека n е такова, че може да се представи като сума на k последователни положителни цели числа, най-малкото от които е m . Тогава

$$n = m + (m+1) + \dots + (m+k-1) = \frac{m + (m+k-1)}{2} \cdot k.$$

Следователно броят на търсените начини е равен на броя на наредените двойки от цели числа (m, k) , които са решения на диофантовото уравнение

$$2n = (2m+k-1)k \text{ и за които } m > 0, k > 1. \text{ Тъй като } 2m = \frac{2n}{k} - k + 1 \text{ и}$$

$k < 2m + k - 1$, то наредената двойка числа (m, k) ще е решение на задачата, ако:

- k дели $2n$;

- $\frac{2n}{k} - k + 1$ е четно число;

- $k^2 < 2n$.

```

#include<iostream>
using namespace std;
int main()
{ int n;
  cin >> n;
  int s=0;
  for(int k=2; k*k<2*n; k++)
    if (!(2*n%k) && !((2*n/k-k+1)%2)) s++;
  cout << s << endl;
  return 0;
}

```

Задача В2. СКОКОВЕ, автор Емил Келеведжиев

По дължината на една алея, на разстояние един метър една от друга, са разположени $N+1$ площадки, номерирани с целите числа от 0 до N , където N е цяло число, $1 < N < 200\,000$. На всяка площадка е поставена кошница с зададен брой ябълки. Скокълъ е известен със способността си да извършва M различни по дължина (измерена в метри) скока, $0 < M < 200$. Скокълъ обича да се разхожда по алеята, като скача от площадка на площадка и събира ябълките които намери в съответната кошница. Разходките му винаги започват от началото на алеята (площадката с номер 0). Когато е на някоя площадка, той може да скочи, с един от възможните M скока, на някои от следващите площадки (такава с по-голям номер). Разходката може да бъде прекратена на произволна площадка до която Скокълъ е достигнал.

Скокълъ може да взема ябълки само от кошниците на площадките, в които е попаднал, включително от тази, от която е тръгнал и тази, в която е прекратил разходката. Напишете програма **SKOK**, която да му помогне да събере максимален брой ябълки.

Вход

Програмата трябва да прочете входните данни от стандартния вход. На първия ред са зададени числата N и M , разделени с интервал. На втория ред са зададени M числа – позволените дължини на скоковете в метри. Всяка от тези дължини е цяло положително число, по-малко от 1000. На третия ред са зададени $N+1$ цели числа – бройките на ябълките, подредени в нарастващ ред на номерата на съответните им площадки. На една площадка може да има най-малко нула и най-много 1000 ябълки. Всички числа във втория и третия ред на входа са разделени с по един интервал.

Изход

Програмата трябва да изведе на един ред в стандартния изход две цели числа, разделени с един интервал. Първото от тези числа трябва да е равно на максималния брой ябълки, които Скокълъ може да събере по време на разходката, а второто число да е равно на номера на последната площадка, до която той е достигнал, получавайки този максимален брой. Ако са възможни