

```

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