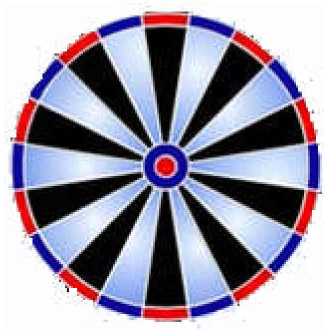
Problema 2 - dartz — 100 puncte



Alex este mare jucător de dartz. Pentru acest joc se foloseşte o tablă în formă de disc împărțită în regiuni, iar fiecare regiune are asociat un punctaj (număr natural nenul). Jocul de desfăşoară în patru etape. La fiecare etapă jucătorul are la dispoziție trei săgeți pe care le aruncă spre tablă. Punctajul obținut de fiecare săgeată este egal cu punctajul regiunii în care a ajuns acesta. Punctajul total **S** obținut de jucător la finalul jocului este suma punctajelor de la cele patru etape.
Neavând partener de joc, Alex se hotărăşte să joace singur toate cele patru etape. Totodată, pentru a lucra şi la aritmetică, el decide ca punctajul fiecărei etape să fie egal cu produsul punctajelor obținute de cele trei săgeți. De exemplu, dacă la o etapă punctajele obținute de cele trei săgeți sunt: **3**, **4**, **5**, atunci punctajul etapei va fi **60** (**60=3·4·5**).
Spre surprinderea lui Alex, după aruncarea săgeților, la fiecare etapă punctajele obținute de cele trei săgeți sunt numere naturale nenule consecutive. În plus, suma punctajelor a două etape este egală cu suma punctajelor celorlalte două etape.

Cerință

Să se scrie un program care să citească punctajul total **S** şi să determine pentru fiecare etapă cel mai mic punctaj pe care poate să-l obțină o săgeată.

Date de intrare

Fişierul de intrare `dartz.in` conține o singură linie pe care este scris numărul natural **S**.

Date de ieşire

Fişierul `dartz.out` va conține o singură linie pe care sunt scrise patru numere naturale nenule **a b c d**, separate prin câte un spațiu, **a** reprezentând cel mai mic punctaj pe care poate să-l obțină o săgeată la prima etapă, **b** reprezentând cel mai mic punctaj pe care poate să-l obțină o săgeată la a doua etapă, **c** reprezentând cel mai mic punctaj pe care poate să-l obțină o săgeată la a treia etapă, **d** reprezentând cel mai mic punctaj pe care poate să-l obțină o săgeată la a patra etapă.

Restricții şi precizări:

- **1 ≤ S ≤ 80000000**
- **S** este număr natural
- Pot exista mai multe soluții. Se cere doar una dintre ele.
- Pentru toate testele utilizate la evaluare există soluție.

Exemplu

dartz.in	dartz.out	Explicație

1560	3 8 3 8	O soluție posibilă poate fi cu punctajele pentru fiecare etapă:
		`etapa 1: 3, 4, 5`
		`etapa 2: 8, 9, 10`
		`etapa 3: 3, 4, 5`
		`etapa 4: 8, 9, 10`
		La etapele `1` şi `3` se obțin punctajele `60`, iar la etapele `2` şi `4` punctajele `720`. Punctajul total este `60+720+60+720=1560`.
		Se observă că restricția ”*suma punctajelor obținute la două etape este egală cu suma punctajelor de la celelalte două etape*” se verifică `(60+720=60+720)`.

Timp maxim de execuție/test: 1 secundă

Descrierea soluției: dartz

prof. Doru Popescu Anastasiu, C.N. "Radu Greceanu" Slatina

Pentru etapa **i** (**i = $\overline{1,4}$**), se obțin punctajele: **a[i]-1**, **a[i]**, **a[i]+1**. Punctajul acestei etape este:

t[i]=(a[i]-1)·a[i]·(a[i]+1)=a[i]^3-a[i].

Fără a restrânge generalitatea, se poate considera că suma punctajelor din primele două etape este egală cu suma punctajelor din ultimele două etape:

t[1]+t[2]=t[3]+t[4]=S/2

1. Un algoritm care obține **100p** este următorul:

O soluție a problemei se poate obține luând **t[3]=t[1]** şi **t[4]=t[2]**. Astfel problema se reduce la găsirea doar a lui **t[1]** şi a lui **t[2]**. Pentru determinarea acestor două valori, se construieşte un vector **x=(x[1],x[2],...,x[k])** unde **x[i]=i·(i-1)·(i+1)**, **i = $\overline{1,n}$**, iar **k** este cel mai mare numar natural cu proprietatea **k·(k-1)·(k+1)≤S**.

Pentru determinarea lui **t[1]** se parcurge vectorul **x** , verificând fiecare **x[i]** dacă valoarea **S/2-x[i]** se găseşte în **x**. În caz afirmativ, căutarea se termină obținându-se o soluție:

t[1]=x[i] şi **t[2]=S/2-x[i]**.

2. Un alt algoritm care obține **50p** este următorul:

```
┌Dacă s nu este divizibil cu 12 atunci nu există soluție.
│ altfel
│┌pentru t[1]=6,…,[S/4] executa //6=1*2*3
││  t[2]=[S/2]-t[1]
││         //verificăm dacă există soluție în etapele 1 şi 2 pentru valorile lui t[1] si t[2]
││           a[1]=1
││    ┌cât timp a[1]^3-a[1]<t[1] execută a[1]=a[1]+1
││    └■
││  a[2]=1
││    ┌cât timp a[2]^3-a[2]<t[2] execută a[2]=a[2]+1
││    └■
││   ┌dacă a[1]^3-a[1]=t[1] si a[2]^3-a[2]=t[2] atunci
││   │ ┌pentru t[3]=6,…, [S/4] executa
││   │ │    t[4]=[S/2]-t[3]
││   │ │       //verificam dacă există soluție în etapele 3 şi 4 pentru valorile lui t[3] şi t[4]
││   │ │    a[3]=1
││   │ │   ┌cât timp a[3]^3-a[3]<t[3] execută a[3]=a[3]+1
││   │ │   └■
││   │ │    a[4]=1
││   │ │   ┌cât timp a[4]^3-a[4]<t[4] execută a[4]=a[4]+1
││   │ │   └■
││   │ │   ┌dacă a[3]^3-a[3]=t[3] si a[4]^3-a[4]=t[4] atunci
││   │ │   │        scrie a[1]-1, a[2]-1, a[3]-1,a[4]-1
││   │ │   │        iesire structuri ciclice
││   │ │   └■
││   │ └■
││   └■
│└■
└■
```