## ЗАДАЧА 1.1. МОНЕТИ

Дадено е множество от M цели положителни стойности и неограничен брой монети за всяка от зададените стойности. Една от стойностите е 1. Да разгледаме следната задача: Зададена сума $S$ да бъде платена с минимален брой от зададените монети.

Известно е, че в някои случаи задачата се решава от следният “лаком” алгоритъм: да намерим най-голямата стойност на монета, която е по-малка или равна на $S$ и да я извадим $S$. Продължаваме така, докато S стане нула. Броят на монетите, използвани за нулирането на S по описания алгоритъм, изглежда минимален.

В много случаи горното твърдение е в сила, но за някои множества от стойности на монетите и за някои $S$ лакомият алгоритъм не намира оптималния брой. Например, за множеството от стойности {1, 2, 5, 7, 10} и S = 14, лакомият алгоритъм намира решение с 3 монети (14=10+2+2), докато очевидното минимално решение е с 2 монети (14 = 7 + 7).

Възниква въпросът – за какви множества от стойности на монетите лакомият алгоритъм не намира оптимално решение. Напишете програма **COINS**, която по зададено множество от стойности на монетите проверява дали има стойност S, която може да бъде платена с по-малък брой монети, отколкото показва лакомият алгоритъм.

Програмата трябва да чете данните от **стандартния вход**. На първия ред ще е зададен броят $M$ на различните стойности на монетите ($1 < M < 100$). На втория ред, разделени с по един интервал, ще бъдат зададени $M$-те стойности $a_1, a_2, \ldots, a_M$ ($1=a_1 < a_2 < \ldots < a_M \leq 7000000$). На третия ред ще бъдат зададени две цели числа $x$ и $y$ ($0 < x < y \leq 7000000$), разделени с един интервал.

Програмата трябва да изведе на първия ред на **стандартния изход** стойност $S$, $x \leq S \leq y$, за която лакомият алгоритъм не успява да намери оптималното решение. На втория ред програмата трябва да изведе числата $b_1, b_2, \ldots, b_M$, разделени с интервали. Числата показват по колко броя монети от съответната стойност са използвани за представянето на $S$. Подреждането е според реда, по които са зададени стойностите на входа, т.е. $S = a_1b_1 + a_2b_2 + \ldots + a_M b_M$. Общият брой използвани монети $b_1+b_2+\ldots+b_M$ трябва да бъде по-малък от броя на монетите, намерен от лакомия алгоритъм.

Входните данни гарантират съществуването на поне едно S, за което лакомия алгоритъм не дава най-доброто решение.

**Пример**

| Вход | Изход |
|---|---|
| 5<br>1 2 5 7 10<br>1 100 | 14<br>0 0 0 2 0 |