

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА Е2. СЪКЛЕНИ ЧАШИ

Очевидно е, че чаша с по-голям диаметър не може да се вложи в чаша с по-малък диаметър, дори и при минимални дебелини на стените ѝ. За да не търсим постоянно кои чаши са с по-големи и кои – с по-малки диаметри, нека подредим чашите в редица в намаляващ ред на диаметрите им. Правим го според следната логика:

- „Набеждаваме“ първата чаша, че е най-голяма (с най-голям диаметър)
- Проверяваме втората дали не е „по-добра“ (с по-голям диаметър) от първата. Ако е така – разменяме местата на първата и втората чаша. От програмистка гледна точка това означава да разменим стойностите на променливите, които описват първата и втората чаша. В тази задача всяка от чашите се описва с по две данни (за диаметър и за дебелина на стената), следователно трябва да разменим стойностите на променливите за диаметрите на чашите и после и стойностите на променливите за дебелини на стените им. Ако втората чаша не е „по-добра“ от първата – нищо не правим. Така след изпълнението на тази стъпка е сигурно, че първа чаша в редицата се е оказала по-добрата от двете.
- Има още една чаша в редицата (третата чаша), която е кандидат да подмени „най-добрата“ чаша. Ако третата е „по-добра“ от първата – разменяме местата им както направихме при предишната стъпка с първата и втората чаши. След тази стъпка е сигурно, че на първо място в редицата се е оказала „най-добрата“ чаша (с най-голям диаметър).
- За останалите две чаши в редицата трябва отново да проверим дали са правилно подредени (втората да е „по-добра“ от третата) и ако това не е така, да им разменим местата (да разменим стойностите на променливите, които ги описват).

След като сме подредили чашите по големина се опитваме да поставим втората чаша в първата, ако това е възможно. Влагането е възможно само ако диаметърът на втората е по-малък или най-много равен на вътрешния диаметър на първата чаша (вътрешният диаметър се получава като разлика на външния ѝ диаметър и удвоената дебелина на стената на чашата). При успешно влагане дължината на редицата вложени чаши е диаметъра на първата чаша. При неуспешно влагане дължината на редицата вложени чаши е сумата от диаметрите на първата и втората. Остава сега да направим опит да вложим и третата чаша в някоя от предишните две. Първо пробваме да я вложим във втората. Ако е възможно, дължината на редицата вложени чаши не се променя. При преценка за влагане на третата чаша в първата трябва да имаме предвид следното важно обстоятелство: първата чаша може да не е празна, защото в нея може да е поставена вече втората. В този случай мястото за влагане на третата е останалото пространство между стените на първата и втората. Проверяваме дали това пространство е достатъчно за влагането на третата чаша и ако не е, единственото което ни остава е да долепим третата чаша към редицата на предишните две (евентуално вложени една в друга), т.е дължината на редицата вложени чаши ще увеличим с диаметъра на третата.

Сорс-кодът на програмата

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

int main() {
    int d1,d2,d3,c1,c2,c3,buff,L;
    cin >> d1 >> c1 >> d2 >> c2 >> d3 >> c3;
    /// Подреждаме кутиите/чашите по големина според външния им размер
```

```

if (d2 > d1) {buff=d2; d2=d1; d1=buff; buff=c2; c2=c1; c1=buff;}
if (d3 > d1) {buff=d3; d3=d1; d1=buff; buff=c3; c3=c1; c1=buff;}
if (d3 > d2) {buff=d3; d3=d2; d2=buff; buff=c3; c3=c2; c2=buff;}
/// Опитваме да вложим втората кутия/чаша в първата
if (d2<=d1-2*c1) {
    L = d1;
    /** ако третата не може да се вложи във втората или в междината м-у първата и
        втората, поставяме я до първата */
    if (!(d3 <= d2-c2*2) or (d3 <= d1-2*c1-d2)) L+=d3;
}
else {
    L = d1+d2; /// първата и втората са поставени една до друга
    /** ако третата не може да се вложи в никоя от предишните две,
        поставяме я до тях в редицата */
    if (!(d3 <= d1-2*c1) or (d3 <= d2-2*c2)) L+=d3;
}
cout << L << endl;
return 0;
}

```

Автор: Евгений Василев