

Задача СУМА

Пояснение към решенията

Бавно решение – файл sum_52p.cpp

Функцията `run()` обработва поотделно всяка поредна тройка числа от входа. Извършва генериране на всички пермутации на цифрите на числото `a` и всяка от тези пермутации се записва във вектора от низове `vector<int> x`. Това става чрез извикване на функцията `make(a, x)`. Дефиницията на функцията `make(a, x)` е следната:

```
void make(int a, vector<int>&x)
{
    string s = to_string(a);
    sort(begin(s), end(s));
    do
    {
        x.push_back(stoi(s));
    } while(next_permutation(begin(s), end(s)));
}
```

Същото се извършва и за числото `b` от входа. Всичките пермутации на цифрите му се записват в последователните елементи на вектора `vector<int> y`. Чрез двоен цикъл се проверява за кои елементи на `x` и `y`, сумата им е равна на `c`.

```
int k=0;
vector<int> v1, v2;
for(int i=0; i<x.size(); i++)
    for(int j=0; j<y.size(); j++)
    {
        if(x[i]+y[j]==c)
        {
            v1.push_back(x[i]); v2.push_back(y[j]);
            k++;
        }
    }
```

В горния фрагмент в променливата `k` се получава броя на двойките `x[i], y[j]`, за които `x[i]+y[j]==c`. Стойностите на тези двойки се записват като елементи на векторите `v1` и `v2`. Накрая се отпечатват:

```
cout << k << endl;
for(int i=0; i<k; i++) cout << v1[i] << " " << v2[i] << endl;
```

Бързо решение – файл sum_100p.cpp

За да направим решение за 100 точки, трябва да премахнем двойния цикъл в предното решение. Генерираме всички пермутации на цифрите само за числото `a`. Не е нужно да използваме вектор за съхраняване на всичките пермутации. При получаване на поредната пермутация в низа `x` използваме стойността `c-a`, преобразуваме я в низ `string z=to_string(c-a)` и трябва да проверим дали низовете `string y = to_string(b)` и `z` съдържат едни и същи елементи. За целта сортираме двата низа:

```
sort(begin(y), end(y));  
sort(begin(z), end(z));
```

и проверяваме дали $z==y$. Ако са равни, записваме числата a и $c-a$ във векторите $v1$ и $v2$ и увеличаваме брояча k :

```
v1.push_back(a); v2.push_back(c-a);  
k++;
```

Накрая отпечатваме:

```
cout << k << endl;  
for(int i=0; i<k; i++) cout << v1[i] << " " << v2[i] << endl;
```

Зорница Дженкова