

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ЧИСЛОВА РЕДИЦА

Тази задача може да бъде решена по няколко начина. Няма да коментираме частта и за определяне на най-дългата последователност от съседни равни елементи – това е класическа задача за броене, която би трябвало всички ученици да могат да направят. По-трудна е втората част. Най-очевидното решение е елементите от редицата да се прочетат в масив, да се ходи последователно по всички четворки от съседни елементи и за всяка да се проверява (с няколко вложени if-а) дали се изпълнява условието четирите съседни елемента да са различни. Това решение крие най-много опасности от грешки, поради вложените if-ове, съчетано със съобразяването на индексите на елементите. Освен това, то е най-малко „универсално”. Друго решение може да се получи, ако се използва допълнителен масив с 8 елемента, преди началото на разглеждането на поредната четворка той се занулява и, когато се разглежда поредният елемент от дадена четворка, неговата стойност да се използва като индекс, за да се сложи в допълнителния масив индикация, че съответната стойност се е срещнала в четворката. След това да се преброи колко различни стойности са се срещнали в четворката. Това решение дава по-универсален подход за решаването на задача от типа: *дадена е редица от N числа, чийто стойности са в диапазона от 1 до K ($K < N$). Определете колко на брой са подредиците от K съседни елемента, в които се срещат всички стойности от 1 до K .* Тук, с използване на допълнителен масив, може да се построи решение, което е линейно по N . За целта допълнителният масив не трябва да се занулява преди всяка следваща подредица от K съседни елемента, а в него трябва да брой коя стойност колко пъти се е срещнала и колко са различните стойности в поредната подредица от K елемента. Това може да става линейно като при преминаването към следващ елемент се извършват действия, свързани с премахването на един елемент и добавянето на нов (какви точно – съобразете сами:-).

Още един начин да се реши задачата е да се съобрази, че $1+2+4+8=15$ и това е единствения случай, когато сумата на четири съседни елемента от масива дава такава стойност. Тогава трябва просто да се поддържа сумата от четирите съседни елемента, когато се преминава към следващ елемент в редицата (т.е. да се вади стойността на елемента, който отпада и да се добавя стойността на новия), да се проверява нейната стойност и, ако е 15, да се добавя 1 към брояча на четворките с търсеното свойство.

Всъщност за решаването и на двете подточки от задачата няма нужда от масив, а само от четири променливи, в които да се намират стойностите на последната четворка съседни елементи.

Реализацията по-долу използва идеята със сума, равна на 15 и не използва масив.

```

#include <iostream>
using namespace std;

int n;

int main()
{
    int a=100;
    int b=200;
    int c=300;
    int d=400;
    int s=1000;
    int i;
    int qcount=0;
    int maxeqcount=0;
    int curreqcount=0;
    cin >> n;
    for (i=1;i<=n;i++)
    {
        s=s-a;
        a=b;b=c;c=d;cin >> d;
        s=s+d;
        if (s==15) qcount++;
        if (c==d)
            curreqcount++;
        else
            curreqcount=1;
        if (curreqcount>maxeqcount)
            maxeqcount=curreqcount;
    }
    cout << maxeqcount << " " << qcount << endl;
}

```

Автор: Руско Шиков