

Задача ЧЕТНО ПРОИЗВЕДЕНИЕ

Пояснение към решението

Бавно решение

Прочитаме в масива `a[]` дадената редица. С двоен цикъл изчерпваме всички възможни граници на отрезки, започващи от индекс i и завършващи на индекс j . При всеки фиксиран ляв край i , зареждаме p с единица и за отрезите с десен край j натрупваме в p произведението на числата. Когато то е четно, увеличаваме с единица брояча c .

```
int c = 0;
for (int i = 0; i < n; i++)
{
    int p = 1;
    for (int j = i; j < n; j++)
    {
        p *= a[j];
        if (p % 2 == 0) c++;
    }
}
cout << c << endl;
```

По-бързо решение

Както в бавното решение използваме двоен цикъл, но не пресмятаме произведенията в отрезите, а следим кога за първи път ще се появи четно число от редицата. Тогава текущото произведение и всички следващи произведения ще бъдат четни, и техният брой е $n - j$. Така увеличаваме брояча c с $n - j$ и излизаме от цикъла.

```
long long int c = 0;
for (int i = 0; i < n; i++)
{
    for (int j = i; j < n; j++)
        if (a[j] % 2 == 0)
            {c = c + n - j; break;}
}
cout << c << endl;
```

Подходящото излизането от цикъла намалява времето за работа на програмата, но в случаите, когато има много на брой само нечетни числа в началото на отрезите, тази програма ще работи бавно.

Бързо решение

Намираме броя t на всички отрезки. Това може да стане например със следния цикъл:

```
long long int t=0;
int nn=n;
for(int i=0;i<n;i++) {t +=nn; nn--;}

```

След това намираме броя $t1$ на отрезите, които имат нечетно произведение на елементите си. Това може да стане чрез единичен цикъл:

```
long long int t1 = 0;
long long int c = 0;
for (int i = 0; i < n; i++)
    if (a[i] % 2 == 0) c=0;
    else {c++; t1 += c;}

```

Накрая отпечатваме разликата `cout << (t - t1) << endl;`

Бързо решение с малко памет

Забелязваме, че в единичния цикъл на бързото решение използваме последователно и еднократно стойностите от масива `a[]`. Това означава, че може да нямаме масив в програмата, а да четем от входа тези последователни стойности. Така липсата на голям масив значително намалява паметта:

```
for (int i = 0; i < n; i++)
{
    int a; cin >> a;
    if (a % 2 == 0) c=0;
    else {c++; t1 += c;}
}
cout << (t - t1) << endl;

```

Емил Келеведжиев