

## АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА СПОРТ

Последователно намираме начините на класиране като генерираме всичките пермутации (размествания) на въведените букви. Намираме дължината на въведения низ  $n$  - броят на състезателите. Пермутациите на числата от 1 до  $n$  се генерират като се използва масив. С цикъл се обхождат елементите на масива:

```
c=1;
for (j=1; j<=n; j++)
{a[j]+=c;
  if(a[j]>m){a[j]=0;c=1;}
  else c=0;
}
```

За елемент с номер едно се извършва увеличаване на стойността му с единица, защото първоначално променливата  $c$  има стойност 1. Когато стойността на някой елемент от масива надмине  $m$ , променливата  $c$  получава отново стойност 1, с цел да бъде прибавена тази стойност към следващия елемент. Когато при излизане от цикъла **for** стойността на  $c$  не е нула, това ни дава информация, че всички клетки в масива са приели максимално възможната си стойност.

В програмата е включен предишния фрагмент, който се извиква многократно чрез оператора **do...while**, докато всички клетки на масива  $a$  се заредят с максималната си стойност. Първоначално масива  $a$  е нулиран, а при всяка промяна на състоянието му неговите стойности стават индекси на масива от символи  $u$  - поредна буква от началния масив от символи. Преди отпечатване проверяваме дали има повтарящи се стойности в масива и при наличие на такива, не се извеждат съответните стойности. Проверката се извършва с функцията **p(a, n)**. Освен това трябва да вземем  $m$  да е равно на  $n-1$ .

```
bool p(int a[], int n)
{ for(int i=1; i<n; i++)
  for(int j=i+1; j<=n; j++)
    if(a[i]==a[j]) return 1;
  return 0;
}
int a[10];
char u[10];
int main()
{
  int n,m;
  cin>>u;
  n = strlen(u);
  m=n-1;
  int c, j;

do
{if(!p(a,n))
{for(j=n; j>=1; j--) cout<<u[a[j]];
cout<<endl; }
c=1;
for(j=1; j<=n; j++)
{a[j]+=c;
```

```
    if(a[j]>m){a[j]=0;c=1;}  
    else c=0;  
  }  
}while(c==0);  
}
```

*Автор: Зорница Дженкова*