

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ЧИСЛО-ПАЛИНДРОМ

Най-лесният начин за решаване на тази задача е обхождане на основите на бройните системи от 2 до 36 и проверка за всяка от тях, явява ли се представянето на даденото число в съответната бройна система *палиндром*.

За проверка, явява ли се един низ *палиндром*, е удобно да се използва следната функция:

```
bool isPalindrome(string s)
{int i,l;
  bool result=true;
  l=s.length();
  for (i=0;i<l;i++)
    if (s[i]!=s[l-1-i]) result=false;
  return result;
}
```

По нататък е необходимо да се знае как се превръща едно десетично число в бройна система с основа k . За това също е удобно да се напише функция, осъществяваща преобразуването на числото n в бройна система с основа k .

```
string convert(int n,int k)
{int i,l,c;
  char digit;
  string result="";
  while (n>0)
    {c=n%k;
     if (c<10) digit=char(int('0')+c);
     else digit=char(int('a')+c-10);
     n=n/k;
     result=digit+result;
    }
  return result;
}
```

По такъв начин, за конкретна основа k на бройната система задачата за проверка на това, явява ли се представянето на числото n в тази бройна система *палиндром*, е решена. Сега ще обходим всички стойности на k от 2 до 36 и за всяка от тях отбелязваме в масива `good[37]` от тип `bool`, явява ли се представянето на числото n в съответната бройна система *палиндром*.

Тази идея се реализира със следния фрагмент от програмата:

```
for (i=2; i<37; i++)
  good[i]=isPalindrome(convert(n,i));
```

За отпечатване на резултата, трябва да намерим броя на елементите от масива `good[37]`, които са равни на `true`. След това е необходимо да се изведе съответната

дума (unique, multiple, none), и при необходимост всички “хубави” основи в нарастващ ред.

Това може да стане например така:

```
for (i=2; i<37; i++)
    if (good[i]) cnt++;
if (cnt==0) cout<<"none";
else
    if (cnt==1) cout<<"unique\n";
    else cout<<"multiple\n";
for (i=2; i<37; i++)
    if(good[i]) cout<<i<<" ";
cout<<endl;
```

Автор: Кинка Кирилова-Лупанова