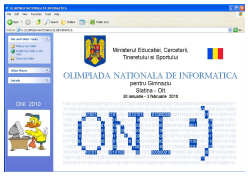


Ministerul Educaţiei, Cercetării, Tineretului şi Sportului
Inspectoratul Şcolar al Judeţului OLT
Colegiul Naţional "Radu Greceanu", Slatina, 31ianuarie 2010

Clasa a V-a

Problema 1 – cluburi

autor prof.Gălăţan Suzana – Inspectoratul Şcolar Judeţean Bistriţa-Năsăud

Descrierea soluţiei

Numărul de cluburi care se formează este egal cu suma resturilor obţinute prin împărţirea repetată a numărului n la 2, atât timp cât n este diferit de 0.

```
┌cât timp n ≠ 0
| ┌dacă n % 2  = 1 atunci
| |  nrclub ← nrclub + 1
| └▪
|  n ← n / 2
└▪
```

Numărul de membri ai fiecărui club este egal cu 2k-1 , unde k este egal cu numărul de împărţiri la 2 al lui n şi la acel pas n%2 este egal cu 1.

```
┌cât timp n ≠ 0
| ┌dacă n % 2  = 0 atunci
| |  k← k*2
| |  n ← n/2
| | ┌altfel
| | | scrie k
| | | k ← k*2
| | | n ← n/2
| | └▪
| └▪
└▪
```

Exemplu: Pentru n = 13 se obţine:

r	**1**	**0**	**1**	**1**
2^{k-1}	2^0	2^1	2^2	2^3
Număr membri	**1**	**-**	**4**	**8**

```
#include <fstream.h>
ifstream fin("cluburi.in");
ofstream fout("cluburi.out");
int main()
{ int N;
  fin >> N ;
  int k = 0,  c = N, cant = 1;
  int x = N;
  while(x)
    { if(x % 2 == 1)k++;
      x = x/2;
     }
  fout << k <<"\n";
  while(c)
    { if(c % 2 == 0)
              c = c/2, cant = cant * 2;
       else
        { c = c/2;
          fout << cant <<" ";
           cant = cant * 2;
        }
    }
     fout << "\n";
    fin.close();
    fout.close();
    return 0;}
```

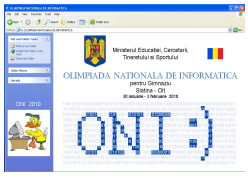
Problema 2 - domino

autor prof. Adriana Simulescu, Liceul "Grigore Moisil" Timişoara

Descrierea soluţiei

Soluţia propusă memorează în variabila ***dist*** poziţia maximă la care acţionează o piesă care a căzut, în variabila ***nd*** numărul de piese doborâte la o atingere şi în ***nmax*** numărul maxim de piese doborâte la o atingere.
Pentru fiecare piesă de domino se verifică dacă este doborâtă de o piesă anterioară, se actualizează numărul de piese doborâte la atingerea curentă şi poziţia până la care se doboară în continuare alte piese.

```
#include<fstream.h>
int n,nd,dmax,nmax,nr,dist,h,p;
ifstream f("domino.in");
ofstream g("domino.ok");
int main()
{ int i;
  f>>n;
  dist=0;
  nd=0;
  for(i=1;i<=n;i++)
    {f>>p>>h;
     if(p>dist)
      {nd++;
       if(nr>nmax) nmax=nr;
       dist=p+h;
       nr=1;
       }
      else{nr++;
         if(dist< p+h )
              dist=p+h;
         }
      }
  if(nr>nmax) nmax=nr;
  g<<nd<<" "<<nmax << '\n';
  g.close();
return 0;
}
```



Problema 3 – max

autor prof.Carmen Mincă, Liceul Teoretic "Ion Neculce" - Bucureşti

Descriere soluţie

O soluţie posibilă se poate obţine prin utilizarea a doi vectori **v** şi **p**, fiecare având maxim **3500** de componente întregi. În vectorul **v** se vor memora numerele din şirul obţinut în urma transformării **T1**, rezultate din numerele din fişierul de intrare. Acestea pot fi citite succesiv într-o variabilă **x**. Pentru fiecare număr memorat în **x** se separă cifrele, se ordonează descrescător aceste cifre iar numărul obţinut se memorează în **v[i]**. În acelaşi timp se memorează în componenta **p[i]** valoarea **p[i]=10^k** , unde **k** reprezintă numărul de cifre ale lui **v[i]**, **p[i]** fiind utilă în construirea numerelor rezultate prin alipirea a două valori din vectorul **v**.

În timp ce se aplică **T1**, se determină şi valoarea maximă din şirul obţinut

Pentru a obţine şirul maximizat rezultat din **T2**, se poate utiliza un algoritm de sortare, de exemplu sortarea prin selectarea maximului. Pentru fiecare **i=1,..,n-1** şi **j=i+1..,n** se construiesc cele două numere rezultate prin prin alipirea, în această ordine, a numerelor **v[i]** şi **v[j]**, respectiv **v[j]** şi **v[i]**, adică **v[i]*p[j] +v[j]** şi **v[j]*p[i]+v[i]**. Se selectează maximul dintre aceste valori.

Numărul **x** cerut se obţine prin alipirea tuturor numerelor din şirul maximizat. Acest lucru se realizează prin scrierea în fişierul de ieşire a tuturor numerelor din şirul maximizat, în ordinea din acest şir, fără spaţii între numere.

```
#include <fstream.h>
int main()
{ int n, v[3501], p[3501];long d,e;
  int i, max=0,x,a,b,c,j;
  ifstream f("max.in");
  ofstream g("max.out");
  f>>n;
  for(i=1;i<=n;i++)
  {   f>>x; v[i]=x;
       if(x<10) p[i]=10;
         else if(x<100)
          { a=x/10; b=x%10; p[i]=100;
            if (a<b)v[i]=b*10+a;
          }
          else
          { a=x/100; x=x%100;
            b=x/10; c=x%10;p[i]=1000;
            if(a<b) { x=a; a=b; b=x;}
            if(a<c) { x=a; a=c; c=x;}
            if(b<c) { x=b; b=c; c=x;}
            v[i]=100*a+10*b+c;
          }
         if(max<v[i]) max=v[i];
    }
    g<<max<<endl;
    for(i=1;i<n;i++)
    for(j=i+1;j<=n;j++)
    { d=v[i];d=d*p[j]+v[j]; e=v[j]; e=e*p[i]+v[i];
      if (d<e) {x=v[i]; v[i]=v[j]; v[j]=x;
                x=p[i]; p[i]=p[j]; p[j]=x;}
    }
    for(i=1;i<=n;i++) g<<v[i];
    g.close();
return 0;}
```

***Descriere soluţie** – prof. Cristina Sichim, C.N. "Ferdinand I" Bacău*

O altă soluţie posibilă pentru rezolvarea cerinţei b) utilizează un vector caracteristic v cu 999 de elemente.

Pentru fiecare cifră x (de la 9 la 1) se afişează mai întâi cifra de un număr de ori egal cu valoarea v[x] din vectorul caracteristic. În continuare, în ordine descrescătoare se afişează numerele y cu două cifre care au prima cifră x, după fiecare dintre ele afişându-se, în ordine descrescătoarele numerele cu trei cifre ce au ca prefix numărul y.

```
#include <fstream.h>
int n,v[1001],i,x,aux,max,y,z,a,b,c;

int main()
{ifstream f("max.in");
 ofstream g("max.out");
 f>>n;
 for(i=1;i<=n;i++)
 { f>>x;
   if(x>9 && x<100 && x/10<x%10) x=x%10*10+x/10;
   if(x>99){a=x%10;b=x/10%10;c=x/100;
                          if(a<b) {aux=a;a=b;b=aux;}
                                   if(a<c) {aux=a;a=c;c=aux;}
                                   if(b<c) {aux=b;b=c;c=aux;}
                                   x=a*100+b*10+c;
                                  }
  v[x]++;
  if(max<x) max=x;
 }
 g<<max<<'\n';
```

//b

```
 for(x=9;x>0;x--)
 { //toate care incep cu cifra x
   while(v[x]--) g<<x;  //cele de o cifra
   for(y=x*10+9;y>=x*10;y--)
       {
       while(v[y]--)g<<y;
       for(z=y*10+x;z>=y*10;z--) while(v[z]--)g<<z;
       }
 }
g<<'\n';
f.close();g.close();
return 0;
}
```