**76)**

int UCP(int n)

{

do

{

if(n % 2 == 0) return n % 10;

n = n / 10;

} while (n);

return -1;

}

**800)**

**Sol 1**

void perfect(int a,int b)

{

int ok=0,i,j,s;

for(i=b;i>=a;i--)

{

s=0; // s=suma divizorilor lui i fara el insusi

for(j=1;j<=i/2;j++)

if(i%j==0)

s+=j;

if(i==s)

{

cout<<i<<" ";

ok=1;

}

}

if(!ok)cout<<"nu exista";

}

**Sol 2**

void perfect(int a,int b)

{

int ok=0,i,j,s;

for(i=b;i>=a;i--)

{

s=0; // s=suma divizorilor lui i fara el insusi

for(j=1; j\*j<i ;j++)

if(i%j==0)

s+=j+i/j;

if(j\*j==i) s+=j;

if(2\*i==s)

{

cout<<i<<" ";

ok=1;

}

}

if(!ok)cout<<"nu exista";

}

**3119)**

n=2374

c=4 => cn=4, p=10

c=7 => c=6, cn=64=6\*10+4, p=100

c=3 => c=2, cn=264=2\*100+64, p=1000

c=2 => cn=2264=2\*1000+264, p=10000

* cn=c\*p+cn

void Impare(int &n)

{

int c, p=1, cn=0;

while(n>0)

{

c=n%10;

if(c%2==1) c--;

cn=c\*p+cn;

p=p\*10;

n/=10;

}

n=cn;

}

**36)**

**Sol 1**

bool prim(int n)

{ int d;

if(n<2) return 0;

for(d=2;d\*d<=n;d++)

if(n%d==0) return 0;

return 1;

}

int i\_prim(int n)

{ int p1,p2;

p1=p2=n;

while(!prim(p1)) p1--;

while(!prim(p2)) p2++;

return p2-p1;

}

**Sol 2**

int i\_prim (int n)

{

int i,p1,p2,ok;

p1=n;

do

{ok=1; // verific daca p1 este numar prim

//if(p1<2) ok=0;

for(i=2; i\*i<=p1; i++)

if(p1%i==0)

{ok=0; break;}

if(!ok) p1--;

} while(!ok);

p2=n;

do

{ok=1; // verific daca p2 este numar prim

//if(p2<2) ok=0;

for(i=2; i\*i<=p2; i++)

if(p2%i==0)

{ok=0; break;}

if(!ok) p2++;

} while(!ok);

return p2-p1;

}

**2859)**

**Sol 1**

bool TreiCifImp(int x)

{

int secv = 0;

while(x)

{

if(x % 2 != 0) secv++;

else secv = 0;

if(secv == 3) return 1;

x = x / 10;

}

return 0;

}

**Sol 2**

bool TreiCifImp(int x)

{

while(x>99)

{ if (x%2==1 && x/10%2==1 && x/100%2==1) return 1;

x = x / 10;

}

return 0;

}

**26)**

**Ex1.**

**N = 6032260 => x= 2002366 // sol. -> vector frevente cifre**

**V[0]=2**

**V[2]=2**

**V[3]=1**

**V[6]=2**

**Ex2. n=280008223 => x=200022388**

long cmmnr(long n)

{int v[10],x=0,i;

**for(i=0;i<10;i++) v[i]=0;**

 while(n)

   { v[n%10]++;

     n=n/10;

   }

for(**i=1;i**<10;i++)  // cea mai mica cifra nenula a lui n se pune prima in x

if(v[i])

{

x=i;

v[i]--;

break;

}

for(i=0;i<10;i++)

   while(v[i])

{

x=x\*10+i;

v[i]--;

}

return x;

}

**43)**

double nreal(int x,int y)  // a=234 ->a=0.234

{  **double a=y;**

    while(a>=1) a=a/10;

    return x+a;

}

**1510)**

int cautare( int n, double x[101], double v)

{  int i,p,u,m;

   p=0; u=n-1;

   while(p<=u)

   {

      m=(p+u)/2;

      if( v<x[m]) u=m-1;

      else if (v>x[m]) p=m+1;

              else return m;

   }

  return -1**;**

 }

Ex. de apel:

cout<<cautare(n,x,v);

**2995)**

**n=2973 -> v=(3,7,9,2) -> n=27 92 74**

void inserare(int &n)

{int i,k=0,v[10];

 while(n){v[++k]=n%10;

              n/=10;

              }

for(i=k;i>1;i--)

n=n\*100+v[i]\*10+abs(v[i]-v[i-1]);

n=n\*10+v[1];

}

**67)**

int prim( int q)

{ if(q<2) return 0;

for(int d=2;d\*d<=q;++d)

if(q%d==0) return 0;

return 1;

}

int a\_prim( int n)

{ for(int d=2;d\*d<n;++d)

if(n%d==0 && prim(n/d)) return 1;

return 0;

}

**902)**

int det(int n)

{ int p=1,i,m;

i=1; // p=i! > n

while(p<=n)

{ i++;

p=p\*i;

}

m=p/i; // m=(i-1)! <= n

if(p-n<n-m) return p;

else return m;

}

**2830)**

**Sol1**

int DivImpar(int a, int b)

{int r;

while(b>0)

{r=a%b;

a=b;

b=r;

}

while (a%2==0) a/=2;

return a;

}

**Sol2**

int DivImpar(int a, int b)

{ int d,i;

for (i=1; i<=min(a,b); **i+=2**)

if (a%i==0 && b%i==0) d=i;

return d;

}

**912)**

int prim(int n)

{

int i;

for(i = 2; i\*i <= n; i++)

if(n % i == 0)

return 0;

return 1;

}

void sub(int n, int &a, int &b)

{

a=n-1;

while (!prim(a)) a--;

b=n+1;

while (!prim(b)) b++;

}