

CORRECTION SUJET 26 MAI 2014
8H

```
program S26_05_2014_8h;
```

```
uses wincrt;
```

```
var
```

```
  n,m:integer;
```

```
      {saisie de n et m}
```

```
procedure saisie (var n,m:integer);
```

```
begin
```

```
  repeat
```

```
    writeln('donner les entiers n et m:');
```

```
    readln(n,m);
```

```
  until (n>=100) and (n<m);
```

```
end;
```

```
{fonction qui vérifie si un entier n est premier ou non}
```

```
function premier (n:integer):boolean;
```

```
var
```

```
  nb,i:integer;
```

```
  ok:boolean;
```

```
begin
```

```
  nb:=0;
```

```
  for i:=1 to n do
```

```
    if (n mod i=0) then
```

```
      nb:=nb+1;
```

```
    if nb=2 then
```

```
      ok:=true
```

```
    else
```

```
      ok:=false;
```

```
  premier:=ok;
```

```
end;
```

```
{fonction qui retourne la somme des chiffres d'un entier}
```

```
function somme (n:integer):integer;
```

```
var
```

```
  s,i,a,e:integer;
```

```
  ch:string;
```

```
begin
```

```
  str(n,ch);
```

```
  s:=0;
```

```
  for i:=1 to length(ch) do
```

```
    begin
```

```
      val(ch[i],a,e);
```

```
      s:=s+a;
```

```
    end;
```

```
  somme:=s;
```

```
end;
```

```
{fonction qui vérifie si un entier n est harshad ou non}
```

```
function harsh (n:integer):boolean;
```

```
begin
```

```
  if (n mod somme(n)=0) and  
  (premier(Pred(n))) then
```

```
    harsh:=true
```

```
  else
```

```
    harsh:=false;
```

```
end;
```

```
{procédure qui affiche tous les entiers harshad qui existent entre deux entiers n et m}
```

```
procedure affichage (n,m:integer);
```

```
var
```

```
  i:integer;
```

```
begin
```

```
  for i:=n to m do
```

```
    if harsh(i) then
```

```
      writeln(i, ' car ',i,' est harshad et son
```

```
      prédécesseur ',i-1,' est un nombre
```

```
      premier');
```

```
end;
```

```
{programme principal}
```

```
begin
```

```
  saisie(n,m);
```

```
  affichage(n,m);
```

```
end.
```

**CORRECTION SUJET 26 MAI 2014
9H30**

```
program S26_05_2014_9h30;
uses wincrt;
var
  nb:integer;

  {saisie de nb}
procedure saisie (var nb:integer);
begin
  repeat
    writeln('donner un entier');
    readln(nb);
  until (nb>=4) and (nb mod 2 = 0);
end;

{fonction qui vérifie si un entier n est premier ou
non}
function premier (n:integer):boolean;
var
  nb,i:integer;
  ok:boolean;
begin
  nb:=0;
  for i:=1 to n do
    if (n mod i=0) then
      nb:=nb+1;
    if nb=2 then
      ok:=true
    else
      ok:=false;
  premier:=ok;
end;

{Procédure qui affiche tous les couples de
nombres premiers tels que leur somme est
égale à nb}

procedure affichage ( nb:integer);
var
  i:integer;
begin
  for i:=2 to nb-2 do
    if premier(i) and premier(nb-i) then
      writeln(('i,',nb-i,));
end;
```

```
{programme principal}
begin
  saisie(nb);
  affichage(nb);
end.
```

CORRECTION SUJET 26 MAI 2014
11H

```
program S26_05_2014_11h;  
uses wincrt;  
type  
  tab1=array[1..30] of integer;  
  tab2=array[1..99] of integer;  
var  
  T:tab1;  
  TI:tab2;  
  n:integer;  
  
  {Saisie de n}  
procedure saisie(var n:integer);  
begin  
  repeat  
    writeln('donner la taille du tableau:');  
    readln(n);  
  until (n in [5..30]);  
end;
```

*{Procédure qui remplit un tableau d'éléments
distincts}*

```
procedure remplissage(var  
  t :tab1 ;n :integer) ;  
var  
  i,j:integer;  
  ok:boolean;  
begin  
  repeat  
    writeln('donner l'element 1');  
    readln(T[1]);  
  until (T[1] in [1..99]);  
  for i:=2 to n do  
    repeat  
      writeln('donner l'element:',i);  
      readln(T[i]);  
      j:=i;  
      ok:=false;  
      repeat  
        j:=j-1;  
        if T[i]=T[j] then  
          ok:=true;  
        until (ok=true) or (j=1);  
      until(ok = false) and (T[i] in [1..99]);  
    end;
```

*{Procédure qui permet de trier T selon le principe
décrit}*

```
procedure tri (var T:tab1; var TI:tab2;  
n:integer);  
var  
  i,j:integer;  
begin  
  {Initialisation du tableau TI (toutes les cases à  
  0)}  
  for i:=1 to 99 do  
    TI[i]:=0;  
  {Remplissage du tableau TI tel que chaque case  
  d'indice T[i] contient l'élément T[i]}  
  for i:=1 to n do  
    TI[T[i]]:=T[i];  
  {Remplissage du tableau T avec les éléments  
  différents de 0 de TI}  
  j:=0;  
  for i:=1 to 99 do  
    begin  
      if TI[i]<>0 then  
        begin  
          j:=j+1;  
          T[j]:=TI[i];  
        end;  
      end;  
    end;
```

{procédure qui affiche le tableau T }

```
procedure affichage (t:tab1;n:integer);  
var  
  i:integer;  
begin  
  for i:=1 to n do  
    writeln(T[i]);  
  end;
```

{programme principal }

```
begin  
  saisie(n);  
  remplissage(T,n);  
  tri(T, TI,n);  
  affichage(T,n);  
end.
```

CORRECTION SUJET 26 MAI 2014
14H

```
program sujet14h;
uses winCRT;
type
    tab=array[1..25] of integer;
var
    Nombre,TD,TA:Tab;
    n,j,k:integer;

    {saisie de n}
procedure saisie(var n:integer);
begin
    repeat
        writeln('donner la taille du
    tableau');
        readln(n);
        until (n in [5..25]);
end;

    {procédure qui remplit le tableau
    Nombre avec des éléments strictement
    positifs}
procedure remplissage(var
    Nombre:tab; n:integer);
var
    i:integer;
begin
    for i:=1 to n do
        repeat
            writeln('donner l'element ', i);
            readln(Nombre[i]);
            until(Nombre[i]>0);
end;

    {fonction qui calcul la somme de
    diviseur d'un entier x sauf lui même}
function somme_diviseur
    (x:integer):integer;
var
    s,i:integer;
begin
    s:=0;
    for i:=1 to (x div 2) do
        if x mod i = 0 then
            s:=s+i;
        somme_diviseur:=s;
end ;
```

```
    {Procédure qui remplit les deux
    Tableaux TD et TA (j et k seront les
    tailles respectives de TD et TA)}
Procedure transfert (Nombre:tab;var
TD,TA:tab;n:integer; var j,k:integer;
var
    i:integer;
begin
    j:=0;
    k:= 0;
    for i:=1 to n do
        if somme_diviseur( Nombre[i])>
    Nombre[i] then
        begin
            j:=j+1;
            td[j]:=Nombre[i];
        end
        else if somme_diviseur(Nombre[i])<
    Nombre[i] then
        begin
            k:=k+1;
            ta[k]:=Nombre[i];
        end
    end;

    {Procédure qui permet d'afficher un
    tableau}
procedure affichage(T:tab;
n:integer);
var
    i:integer;
begin
    for i:=1 to n do
        write (T[i], ' ');
    end;

    {Programme principal}
begin
    saisie(n);
    remplissage(Nombre,n);
    transfert(Nombre,TD,TA,n,j,k);
    affichage(TD,j);
    writeln ;
    affichage(TA,k) ;
end.
```