

Correction du TD n°4

Exercice 1 :

```
PROGRAM multcomplex;
TYPE complex=array[1..2]of real;
VAR a,b,c:complex;
```

```
PROCEDURE affiche(z:complex);
BEGIN
write('(',z[1],'+',z[2], 'i)');
END;
```

```
PROCEDURE rentre(var z:complex);
BEGIN
write('Re(z), Im(z) ? ');
readln(z[1],z[2]);
END;
```

```
BEGIN
rentre(a);
rentre(b);
c[1]:=a[1]*b[1]-a[2]*b[2];
c[2]:=a[1]*b[2]+a[2]*b[1];
affiche(a);
write('*');
affiche(b);
write(' = ');
affiche(c);
readln;
END.
```

Exercice 2 :

```
PROGRAM ajoute_matrice;
CONST n=2;
TYPE matrice=array[1..n,1..n]of real;
VAR a,b,c:matrice;
i,j,k:integer;
s:real;

PROCEDURE affiche(m:matrice);
```

```
BEGIN
for i:=1 to n do
BEGIN
for j:=1 to n do
write(m[i,j], ' ');
writeln;
END;
END;
```

```
PROCEDURE rentre(var m:matrice);
BEGIN
for i:=1 to n do
for j:=1 to n do
BEGIN
write('m[',i,',',j,']? ');
readln(m[i,j]);
END;
END;
```

```
BEGIN
rentre(a);
rentre(b);
for i:=1 to n do
for j:=1 to n do
c[i,j]:=a[i,j]+b[i,j];
affiche(a);
writeln('*');
affiche(b);
writeln(' = ');
affiche(c);
readln;
END.
```

Exercice 3 :

```
program suite;
const t=100
var u:array[1..t] of real;
i,n:integer;
BEGIN
write('n? (<',t,')');
readln(n);
u[1]:=1;
for i:=2 to n do
BEGIN
u[i]:=0;
for j:=1 to i-1 do
u[i]:=u[i]+(i-j)*cos(u[j]);
```

```

    END;
writeln('u_n=', u[n]);
readln;
END.

```

Exercice 4 :

```

c) program lanciers;
const n=1000;
type table=array[1..6]of
integer;
var i:integer;
    tab:table;

PROCEDURE range(var tab:table);
var res:integer;
BEGIN
    res:=random(6)+1;
    tab[res]:=tab[res]+1;
END;

PROCEDURE affiche(tab:table);
var j:integer;
BEGIN
    for j:=1 to 6 do
        writeln(j, ' ', tab[j]);
    END;
BEGIN
    randomize;
    for i:=1 to 6 do
        tab[i]:=0;
    for i:=1 to n do
        range(tab);
    affiche(tab);
    readln;
END.

```

d) Le tableau doit aller de 2 à 12 au lieu de de 1 à 6.
 et res est maintenant :
`res:=random(6)+random(6)+2;`
 Attention, ceci est faux :
`res:=2*random(6)+2;`
 Cela revient à ne lancer qu'un seul dé, et à multiplier
 par deux son chiffre.

Exercice 5 :

```

program polynome;
type poly=array[0..19] of real;
var x,puiss,res:real;
    p:poly;
    i,n:integer;

```

```

PROCEDURE vider(p:poly);
var j:integer;
BEGIN
    for j:=0 to 19 do
        p[j]:=0;
    END;

```

```

PROCEDURE enregistre(var
n:integer;var x:real);
BEGIN
    REPEAT
        writeln('degre du
polynome ? (<20)');
        readln(n)
    UNTIL n<20;
    writeln('x?');
    readln(x);
END;

```

```

BEGIN
vider(p);
enregistre(n,x);
for i:=0 to n do
    BEGIN
        writeln('coeff ',i,'?');
        readln(p[i]);
    END;
puiss:=1;
res:=p[0];
for i:=1 to n do
    BEGIN
        puiss:=puiss*x;
        res:=res+p[i]*puiss;
    END;
writeln('Le résultat est ',res);
readln;
END.

```

Correction du TD n°4 bis

Exercice 1 et 2:

```
program popol;
const t=19;
type polynome=array[0..t]of
real;
var n:integer;
    P:polynome;
    x:real;

function
f(x:real;n:integer):real;
var i:integer;
    p:real;
begin
    p:=1;
    for i:=1 to n do
        p:=p*x;
    f:=p;
end;

procedure vide(var G:polynome);
var i:integer;
begin
    for i:=0 to t do
        G[i]:=0;
    end;

procedure demande_info(var
m:integer;var L:polynome;var
y:real);
var i:integer;
begin
    write('degre du
polynome ? (<t,') ');
    readln(m);
    for i:=0 to m do
        begin
            write('coeff de
degre',i,'? ');
            readln(L[i]);
        end;
end;
```

```
write('reel x ? ');
readln(y);
end;
```

```
function
calc_pol(b:integer;Q:polynome;z:
real):real;
var som:real;
    i:integer;
begin
    som:=Q[0];
    for i:=1 to b do
        som:=som+Q[i]*f(x,i);
    calc_pol:=som;
end;
```

```
function
Hcalc_pol(b:integer;Q:polynome;z
:real):real;
var K:real;
    i:integer;
begin
    if b=0 then Hcalc_pol:=Q[0]
    else
        begin
            K:=Q[b-1]+Q[b]*z;
            for i:=b-2 downto 0 do
                K:=Q[i]+K*x;
                Hcalc_pol:=K;
            end;
        end;
end;
```

```
begin
    vide(P);
    demande_info(n,P,x);
    writeln('res=',Hcalc_pol(n,P,x
));
    readln;
end.
```

Exercice 3 :

```

PROGRAM tri_tableau_bulle;
CONST n=10;
TYPE tab=array[0..n-1]of
integer;
VAR t:tab;
    temp:real;

PROCEDURE remplir(var
t:tab;n:integer);
    VAR i:integer;
    BEGIN
        for i:=0 to n-1 do
            t[i]:=random(101);
        END;

PROCEDURE
affiche(t:tab;n:integer);
    VAR i:integer;
    BEGIN
        for i:=0 to n-1 do
            write(t[i], ' ');
        writeln;
    END;

PROCEDURE tribulle(var
t:tab;integer:n);
    VAR i,j:integer;
        temp:reel;
    BEGIN
        for i:=n-1 downto 2 do
            for j:=0 to i do
                if t[j]>t[j+1] then
                    BEGIN
                        temp:=t[j];
                        t[j]:=t[j+1];
                        t[j+1]:=temp;
                    END;
            END;
        END;

BEGIN
    remplir(t,n);
    affiche(t,n);
    readln;
END.

```

d) Le programme fait $(n-1)+(n-2)+\dots+1$ comparaisons,

avec éventuellement 3 affectations à chaque fois. Ce qui donne un maximum de $2n(n-1)$ opérations, équivalent à $2n^2$ pour n grand et un minimum de $n(n-1)/2$.

Le nombre d'opération est maximal si les éléments du tableau sont triés en ordre inverse, minimal si le tableau est déjà trié.

On ne retient pas le facteur devant le n^2 . On dit que l'algorithme est en n^2 .

Exercice 4 :

```

PROGRAM tri_tableau_bulle;
CONST n=10;
TYPE tab=array[0..n-1]of
integer;
VAR t:tab;
    temp:real;

PROCEDURE remplir...
PROCEDURE affiche...

PROCEDURE tri_insertion(var
t:tab;n:integer);
    VAR i,j,k,temp:integer;
    BEGIN
        for i:=1 to n-1 do
            BEGIN
                j:=i-1;
                while (t[j]>t[i]) and
(j>=0) do
                    j:=j-1;
                temp:=t[i];
                for k:=i-1 downto j+1 do
                    t[k+1]:=t[k];
                t[j+1]:=temp;
            END;
        END;

BEGIN
    remplir(t,n);
    affiche(t,n);
    tri_insertion(t,n);
    affiche(t,n);
    readln;
END.

```

d) Le programme fait environ $2i$ opérations à chaque boucle, donc, n^2 au final. C'est encore un algorithme en n^2 . Mais si le tableau est déjà trié, le nombre d'opérations tombe à quelque chose en n .