

# X Info PC 2004

>

## Question 1

```
> entier:=proc(b,c) 3*b+c end;  
entier := proc(b, c) 3*b + c end proc;  
  
> entier(0,2);entier(2,0);  
2  
6
```

## Question 2

```
> poidsFort:=proc(x) iquo(x,3) end;poidsFaible:=proc(x) irem(x,3) end;  
poidsFort := proc(x) iquo(x, 3) end proc;  
poidsFaible := proc(x) irem(x, 3) end proc;  
  
> verifie:=proc(x) entier(poidsFort(x),poidsFaible(x)) end;  
verifie := proc(x) entier(poidsFort(x), poidsFaible(x)) end proc;  
  
> map(verifie, [$0..8]);  
[0, 1, 2, 3, 4, 5, 6, 7, 8]
```

## Question 3

**+** Pour pouvoir tester on commence par fabriquer des tableaux aléatoires

```
> longueurMotif:=proc(t,i,j,m) local k;  
k:=0; while ((k<m) and (t[i+k]=t[j+k])) do k:=k+1 od; k end;  
longueurMotif:=proc(t, i, j, m)  
local k;  
k := 0; while k < m and t[i + k] = t[j + k] do k := k + 1 end do;  
k;  
end proc;  
  
> longueurMotif(test1,0,8,8);  
0
```

## Question 4

```
> longueurMotifMax:=proc(t,i,j,m) local k,max;  
k:=0;max:=0;while (i+k<j) do  
if longueurMotif(t,i+k,j,m) > max  
then max:=longueurMotif(t,i+k,j,m) fi ;  
k:=k+1; od; max end;  
longueurMotifMax := proc(t, i, j, m)  
local k, max;
```

```

    k := 0; max := 0; while i + k < j do
        if max < longueurMotif(t, i + k, j, m) then max := longueurMotif(t, i + k, j, m) end if;
        k := k + 1;
    end do;
    max;
end proc;

```

```

> affarr(test2);
    2, , 1, , 2, , 0, , 1, , 1, , 0, , 2, , 0, , 1, , 1, , 0, , 2, , 0, , 1, , 1, , 2, , 1, , 2, , 0, , 2, , 2, , 7,
> longueurMotifMax(test2,0,8,8);
    8

```

## Question 5

```

> test2:=array(0..22,[2,1,2,0,1,1,0,2,0,1,1,0,2,0,1,1,2,1,2,0,2,2,7]):af
farr(test2);
    2, , 1, , 2, , 0, , 1, , 1, , 0, , 2, , 0, , 1, , 1, , 0, , 2, , 0, , 1, , 1, , 2, , 1, , 2, , 0, , 2, , 2, , 7,
> motifMax:=proc(t,i,j,m) local k,max,kmax;
    k:=0;max:=0;kmax:=0;
    while (i+k<j) do
        if longueurMotif(t,i+k,j,m)>max
        then kmax:=k;max:=longueurMotif(t,i+k,j,m) fi ;
        k:=k+1; od; (kmax,max,t[i+k+max]) end:
> motifMax(test2,0,8,8);
    3, 8, 2

```

## Question 6

```

> imprimerTriplet:=proc(A,L,C)
    (poidsFort(A),poidsFaible(A),poidsFort(L),poidsFaible(L),C) end;
imprimerTriplet := proc(A, L, C) poidsFort(A), poidsFaible(A), poidsFort(L), poidsFaible(L), C end proc
> imprimerTriplet(8,8,2);
    2, 2, 2, 2, 2

```

## Question 7

```

> test3:=array(0..36,[0,0,0,0,0,0,0,0,0,1,0,2,1,0,2,1,0,1,2,1,0,2,1,0,0,
2,1,0,2,1,0,0,2,1,0,0,x]):affarr(test3);
0, , 0, , 0, , 0, , 0, , 0, , 0, , 0, , 0, , 0, , 1, , 0, , 2, , 1, , 0, , 2, , 1, , 0, , 1, , 2, , 1, , 0, , 2, , 1, , 0, , 0,
, 2, , 1, , 0, , 2, , 1, , 0, , 0, , 2, , 1, , 0, , 0, , x,
> motifMax(test3,0,9,8);
    0, 0, 1
> imprimerTriplet(motifMax(test3,0,9,8));
    0, 0, 0, 0, 1
> imprimerTriplet(motifMax(test3,0,9,8));
    0, 0, 0, 0, 1

```

```

> imprimerTriplet(motifMax(test3,1,10,8));
                                0, 0, 0, 1, 2
> imprimerTriplet(motifMax(test3,3,12,8));
                                2, 0, 1, 2, 1
> imprimerTriplet(motifMax(test3,9,18,8));
                                0, 2, 2, 0, 0
> imprimerTriplet(motifMax(test3,16,25,8));
                                0, 2, 2, 2, 1
> imprimerTriplet(motifMax(test3,25,34,8));
                                1, 2, 0, 2, x

> compresseur:=proc(t)
  local boulot;
  boulot:=proc(d)
    local u,mm;
    mm:=motifMax(test3,d,d+9,8);
    u:=mm[3];
    if u=x
    then imprimerTriplet(op(mm))
    else imprimerTriplet(op(mm),boulot(d+mm[2]+1) fi;
  end;
  boulot(0)
end:

> mm:=motifMax(test3,0,9,8);
                                mm := [0, 0, 1]

> compresseur(test3);
0, 0, 0, 0, 1, 0, 0, 0, 1, 2, 2, 0, 1, 2, 1, 0, 2, 2, 0, 0, 0, 2, 2, 2, 1, 1, 2, 0, 2, x

> ct3:=array(0..29,[0, 0, 0, 0, 1, 0, 0, 0, 1, 2, 2, 0, 1, 2, 1, 0, 2,
2, 0, 0, 0, 2, 2, 2, 1, 1, 2, 0, 2, x]):affarr(ct3);
0, , 0, , 0, , 1, , 0, , 0, , 1, , 2, , 2, , 0, , 1, , 2, , 1, , 0, , 2,
, 2, , 0, , 0, , 2, , 2, , 1, , 1, , 2, , 0, , 2, , x]
, 1, , 2, , 0, , 2, , x,

```

## Question 8

Le texte à renvoyer va être initialisé à 000000000, il sera nommé 'renvoi' de longueur courante 'ici'

quand on lit un triplet (A,L,C) on va ajouter L+1 caractères à renvoi qui seront : L caractères de l'ancien renvoi en partant du (ici-9+A)-ième puis le caractère C

quand C n'est pas 0,1 ou 2 on sait avoir fini

```

> decompresser:=proc(tc)
  local longtc,long,k,newt,ici,p,a,l,c;
  longtc:=op(2,op(2,eval(tc)));long:=9;
  for k from 1 to (1+longtc/5) do
  long:=long+1+entier(tc[5*k-3],tc[5*k-2]) od;
  newt:=array(0..long-1);

```

```
for k from 0 to 8 do newt[k]:=0 od; ici:=8;
for k from 1 to (1+longtc/5) do
```

```
a:=entier(tc[5*k-5],tc[5*k-4]);l:=entier(tc[5*k-3],tc[5*k-2]);c:=tc[5*
k-1];
```

```
if l<>0 then for p from 1 to l do newt[ici+p]:=newt[ici-9+a+p]
od fi; ici:=ici+1;
```

```
ici:=ici+1;newt[ici]:=c;
od;affarr(newt);
```

```
end:
```

```
> decompresser(ct3);
```

```
0, ,0, ,0, ,0, ,0, ,0, ,0, ,0, ,1, ,0, ,2, ,1, ,0, ,2, ,1, ,0, ,1, ,2, ,1, ,0, ,2, ,1, ,0, ,0,
,2, ,1, ,0, ,2, ,1, ,0, ,0, ,2, ,1, ,0, ,0, ,x,
```

```
>
```