PARIS DESCARTES

PROG2 - Programmation impérative

TP 04 – Les procédures et les fonctions

Julien Sopena

Février 2008

Exercice 1 : Sous-programmes (suite)

Question 1

On reprend l'exercice du TD précédent, mais l'on modifie le stockage dans le tableau : les nombres entiers positifs ne sont plus suivie d'une valeur négative. Donner le type **t_tab** correspondant à ce nouveau tableau.

```
const
   TAILLE_TAB = 100;
 type
   t_tab = array [1..TAILLE_TAB] of integer ;
```

Question 2

Puisqu'il n'y a plus de "bouchon" il va falloir, conserver le nombre de valeurs enregistrées dans le tableau. Pour ce faire on commence par modifier le sous-programme **saisir** pour qu'il retourne le nombre cette information.

```
function saisir(var tab  : t_tab)  : integer ;
var
   i  : word ;
begin
   i  := 0 ;
   repeat
      i  := i+1 ;
      write ('entrez un nombre : ');
      readln (tab[i]);
   until (tab[i] < 0) or (i = TAILLE_TAB) ;

   if (tab[i] >= 0) then
```

```
    begin
        tab[i+1]  := -1 ;
        saisir  := i ;
    end
    else
        saisir  := i-1 ;
end ;
```

Question 3

Modifier le sous-programme **max** pour qu'il tienne compte de cette taille.

```
function max(tab :t_tab ; N :integer)  : integer  ;
var
    i,m  : word  ;
begin
    m  := tab[1] ;
    for i  := 2 to N do
        if m < tab[i] then
            m  := tab[i] ;
    max  := m ;
end ;
```

Question 4

Modifier le sous-programme **affiche** pour qu'il tienne compte de cette taille.

```
procedure affiche(tab :t_tab ; N :integer)  ;
var
    i  : integer  ;
begin
    write ('[') ;
    if N > 0 then
    begin
        write (tab[1]) ;
        for i  := 2 to N do
            write (', ',tab[i]) ;
    end ;
    writeln (']') ;
end ;
```

Question 5

Modifier le sous-programme **moyennage** pour qu'il tienne compte de cette taille.

```
procedure moyennage(var tab :t_tab ; N :integer)  ;
var
    oldPred,tmp,i  : integer  ;
begin
    if (tab[1] >= 0) and (tab[2] >= 0) then
```

```
    begin
        oldPred  := tab[1];
        i  := 2  ;
        while (tab[i+1] >= 0) do
        begin
            tmp  := tab[i];
            tab[i]  := (oldPred + tab[i] + tab[i+1] ) div 3;
            oldPred  := tmp;
            inc(i);
        end;
    end;
end;
```

Question 6

Écrire un sous-programme **test** qui vérifie que les valeurs contenues dans le tableau sont comprises entre 1 et 39 et retourne le résultat de ce test.

```
function test (tab :t_tab; N :integer)  : boolean  ;
var
    i  : integer  ;
begin
    i  :=1;
    while ( i <= N ) and (tab[i] > 0) and (tab[i] < 40) do
        i  := i + 1  ;
    test  := (i > N)  ;
end;
```

Question 7

Écrire un sous-programme **dessin** qui affiche à l'écran des lignes de caractères X si les valeurs contenues dans le tableau sont comprises entre 1 et 39. Sur chaque ligne, les lignes étant ordonnées comme les indices du tableau, le nombre de caractères X est égal à la valeur contenue dans le tableau. Exemple : Pour le tableau $[5, 2, 4]$ **dessin** affiche

```
XXXXX
XX
XXXX
```

```
procedure dessin (tab :t_tab; N :integer);
var
    i,j  : integer  ;
begin
    if test(tab,N) then
    begin
        for i :=1 to N do
        begin
            for j :=1 to tab[i] do
                write ('X')  ;
            writeln;
        end
```

```
    end ;
end;
```

Question 8

Écrire un sous-programme **dessinDouble** qui après la même vérification fait un affichage symétrique du dessin. Exemple :

```
XXXXXXXX
  XX
 XXXXXX
```

```
procedure dessinDouble (tab :t_tab; N :integer);
var
   i,j,m  : integer ;
begin
   if test (tab,N) then
   begin
      m  := max(tab,N);
      for i :=1 to N do
      begin
         for j :=1 to m-tab[i] do
            write (' ');
         for j :=1 to 2*tab[i] do
            write ('X');
         writeln;
      end ;
   end ;
end;
```

Question 9

Écrire un programme qui teste les sous-programmes et les déclarations précédents.

```
var
   monTab  : t_tab ;
   longueur  : integer ;  begin
   longueur  := saisir(monTab);
   write('Tableau saisi : '); affiche(monTab);
   writeln;
   writeln('Dessin :');
   dessin(monTab,longueur);
   writeln;
   writeln('Dessin double :');
   dessinDouble(monTab,longueur);
   readln;
end.
```

