Les tableaux à une dimension (vecteurs)

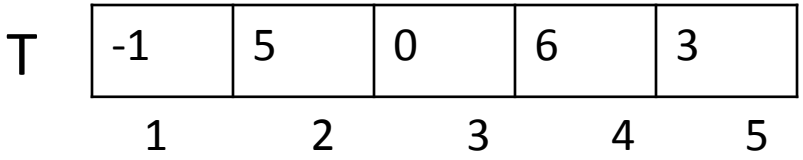
Université A-Mira de Bejaia 2020/2021
Faculté de Technologie
1ere année Technologie
Module : Informatique2
Présenté par : Mme MAMMERI

Définition

Un tableau à une dimension est une structure de données composée d'un nombre fixe d'éléments de même type, rangés dans des cases mémoires contigües.

Exemple :

Soit T un tableau de 5 éléments de type entier



T: est l'identificateur (nom) du tableau

1,2,3,4,5 sont les indices (positions)des éléments du tableau T

-1, 5, 0, 6, 3 : sont les éléments (composantes) du tableau T

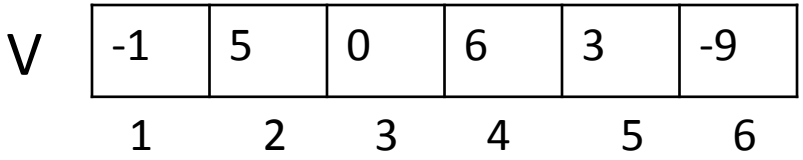
Accéder à un élément d'un tableau

Syntaxe

<Nom du Tableau>[<Indice de l'element>]

Exemple :

Soit V un tableau de 6 éléments de type reel



V[1] : représente le 1er élément du tableau V, V[1]=-1
V[2] : représente le 2eme élément du tableau V, V[2]=5
.
.
V[6] : représente le 6eme élément du tableau V, V[6]=-9

Déclaration d'un tableau

Syntaxe

En algo Variable <id_tableau> :tableau[1..taille_max]de <type> ;

En Pascal **Var** <id_tableau> :array [1..taille_max]**of** <type> ;

Exemple :

En algo : variable T: tableau[1..100]de reel;

En pascal : **var** T: array[1..100]**of** real;

Affecter un élément à un tableau

Syntaxe

En algo <id_tableau>[<indice>] ← élément ;

En Pascal <id_tableau>[<indice>]:= élément ;

Exemple :

En algo T[1]← 4 T[2]← 9

En pascal T[1]:= 4 ; T[2] := 9 ;

Lecture des éléments d'un tableau

En algorithmique :

```
Pour  i  ← 1 à nb_élément faire
  lire([T[i]) ;
Finpour ;
```

En Pascal:

```
for  i := 1 to  nb_élément do
begin
     read(T[i]);
End;
```

Affichage des éléments d'un tableau

En algorithmique :

```
Pour  i  ← 1 à nb_élément faire
  ecrire ([T[i]) ;
Finpour ;
```

En Pascal:

```
 for  i := 1 to  nb_élément do
 begin
      Write (T[i], '    ');
 End;
```

Algorithme/Programme de lecture et d'affichage d'un tableau de n éléments entiers

```
Algorithme lecture_affichage;
 Variables  T: Tableau [1..100] d'entier
i,  n : entier
Debut
    Lire(n)
  Pour i←1 à n faire
     Lire( T[i] )
 FinPour

Pour i←1 à n faire
   ecrire ( T[i], ‘   ‘ )
 FinPour

Fin .
```

```
Program  lecture_Affichage ;
Uses wincrt ;
Var T:array[1..100] of integer;
 i,n: integer;
Begin
read(n) ;
for i:=1 to n do
     read(t[i]);

writeln('les éléménts de T sont :');
 for i:=1 to n do
         write(t[i], '  ');

End.
```

Quelques Algorithmes/programmes sur les tableaux à une dimension

➢ **La somme des éléments d'un tableau T de type réels :**

```
Algorithme somme
Variables T:tableau[1..100] de reel
  i,n:entier
 s:reel
Debut
 Lire(n)
  Pour i←1 à n faire
            Lire( T[i] )
 FinPour
S←0
 Pour i←1 à n faire
     S←s+T[i]
Finpour
Ecrire ('somme=' , s);
Fin.
```

```
Program  Somme ;
Uses wincrt ;
Var  T:array[1..100] of real;
s:real;
 i,n: integer;
Begin
read(n) ;
for i:=1 to n do
        read(t[i]);

s:=0;
for i:= 1 to n do
        s:=s+t[i];

write( 'la somme=  ', s:8:2);
End.
```

Quelques Algorithmes/programmes sur les Tableaux à une dimension

➢La somme de deux tableaux A et B

```
Algorithme somme
Variables  A, B, C:tableau[1..100] de reel
  i,n:entier
Debut
 Lire(n)
  Pour i←1 à n faire
        Lire( A[i] )
 FinPour
 Pour i←1 à n faire
        Lire( B[i] )
 FinPour
Pour i←1 à n faire
   C[i]← A[i]+B[i]
Finpour
 Pour i←1 à n faire
   ecrire( C[i], ‘     ‘ )
 FinPour
Fin.
```

```
Program  Somme ;
Uses wincrt ;
Var A, B, C :array[1..100] of real;
 i,n: integer;
Begin
read(n) ;
for i:=1 to n do
read(A[i]);
for i:=1 to n do
read(B[i]);

for i:= 1 to n do
c[i]:=A[i]+B[i];

for i:=1 to n do
write(C[i]:8:2, '      ');
End.
```

Quelques Algorithmes/programmes sur les Tableaux à une dimension

➢Le produit scalaire de deux tableaux A et B

```
Algorithme produit_scalaire
Variables  A, B :tableau[1..100] de reel
 i,n:entier
   Ps:reel
Debut
 Lire(n)
  Pour i←1 à n faire
       Lire( A[i] )
 FinPour
 Pour i←1 à n faire
       Lire( B[i] )
 FinPour
Ps←0
Pour i←1 à n faire
  ps← ps+A[i]*B[i]
Finpour
 ecrire(  'le produit scalaire =  ' ,  ps  )

Fin.
```

```
Program  produit_scaliare ;
Uses wincrt ;
Var A, B :array[1..100] of real;
i,n: integer;
 ps:real;
Begin
read(n) ;
for i:=1 to n do
read(A[i]);
for i:=1 to n do
read(B[i]);
ps:=0;
for i:= 1 to n do
ps:=ps+A[i]*B[i];
write('produit scalaire = ',  ps);

End.
```

Quelques Algorithmes/programmes sur les Tableaux à une dimension

➢La recherche d'un élément dans un tableau et sa position s'il existe

```
Algorithme recherche
Variables T:tableau[1..50] de reel
  i,n, p:entier
  X:reel
Trouve:booleen
Debut
 Lire(n)
  Pour i←1 à n faire
            Lire( T[i] )
 FinPour
Lire(X)
Trouve←faux
  i←1
Tantque (i≤n) et (trouve=faux) faire
        Si T[i]=X alors
           trouve←vai
             p←i
       finsi
       i←i+1
Fintantque
Si  trouve = vrai  alors
  ecrire (‘ l’’element existe, sa position=‘ , p)
Sinon
ecrire (‘ l’’element n’’existe pas ‘)
Finsi
Fin.
```

```
Program Recherche;
Uses wincrt;
Var  T : Array[1..50] of real;
X : real;
 i ,n, p : integer;
Trouve : boolean ;
Begin
Read (n);
for i:=1 to N do
read (T[i]);
read(X);
trouve:=false;
i:=1 ;
while (i<=n) and (Trouve=false) do
begin
if T[i] = X then
begin
Trouve := true;
p:=i;
end ;
i:=i+1;
end;
if Trouve = true  then
write('L’’element ‘   , X,   ‘ existe sa position = :',  p)
else
write(X,' n''existe pas ');
End.
```

Quelques Algorithmes/programmes sur les Tableaux à une dimension

➢Le nombre d'occurrences d'un élément dans un tableau et ses positions

```
Algorithme  NbOccurence
Variables   T:tableau[1..100] de reel
P: tableau[1..50 ] d'entier
  i,n , nb:entier
  X:reel
Debut
 Lire(n)
  Pour i←1 à n faire
            Lire( T[i] )
 Fin-Pour
Lire(X)
Nb←0
  i←1
Tantque (i≤n)  faire
        Si T[i]=X alors
           nb←nb+1
            P[nb]←i
        finsi
       i←i+1
FinTantque
ecrire (' le nombre d'occurrence de '  ,  X,  ' =' , nb)
ecrire (' ses positions  : ')
Pour i←1 à nb faire
   ecrire (P[i], '    ')
finpour
Fin.
```

```
Program  NbOccurence;
Uses wincrt;
Var T : Array[1..100] of real;
p: array[1..50] of integer;
X : real;
N, I,nb : integer;
Begin
Read (N);
for i:=1 to N do
read (T[i]);
read(X);
nb:=0;
i:=1 ;
while i<=N do
begin
if T[i] =X then
begin
nb:=nb+1;
p[nb]:=i;
end ;
i:=i+1;
end;
writeln('Le nombre d''occurrence s de',X, ' = ', nb);
write(' ses positions :');
for i:=1 to nb do
write (p[i], '  ');
End.
```