بسم الله الرحمن الرحيم

اسأل الله ان اجدكم في احسن حال و الصحة, انتم و ذويكم

السلام عليكم و رحمه الله و بركاته

Salam, j’ espère que vous allez bien ainsi que votre famille

La récursivité

Rappel et exercices

- ✓ Principe
- ✓ Implémentation
- ✓ Exemple
- ✓ Finalité
- ✓ Illustration d'une solution récursive
- ✓ Exercices 1-2-3 série 2
- ➢ développement des solutions des exercices

développement des solutions des exercices

EX3 : Ecrire un sous-programme récursif qui inverse les éléments d'un tableau d'entiers.

- Pour ce nouveau exercice, on verra la manipulation des tableaux

- Les tableaux se prêtent bien à la récursivité dans la plupart des cas

- Nous allons présenter un exemple d'un tableau et nous y appliquons l'inversion demandée.

Exemple: Soit un tableau unidimensionnel (vecteur) de dimension n = 8

-6	40	28	36	-12	5	49	20
0	1	2	3	4	5	6	7

- Rappelons que l'indice du tableau en C, commence :

- Le premier élément a l'indice 0

- Donc le dernier élément à l'indice (n-1)

Solutions de Ex 3

-6	40	28	36	-12	5	49	20
0	1	2	3	4	5	6	7

- Nous voulons inverser les éléments du tableau, donc nous devrons obtenir ce nouveau résultat:

20	49	5	-12	36	28	40	-6
0	1	2	3	4	5	6	7

✓ Donc le premier élément est devenu le dernier et le dernier a pris place du premier

✓ Et le deuxième est devenu l'avant dernier et l'avant dernier est devenu le deuxième.

✓ Et ainsi de suite ….

✓ Il est demandé d'utiliser le même tableau → donc, à la fin du travail, le résultat est contenu dans le même tableau.

✓ Nous n'avons pas besoin d'utiliser un autre tableau

Solutions de Ex 3

✓ Il est important de remarquer que si nous inversons chaque élément du tableau, du premier au dernier, nous obtenons à la fin le tableau lui-même (je vous invite à le faire pour voir).

-6	40	28	36	-12	5	49	20
0	1	2	3	4	5	6	7

20	40	28	36	-12	5	49	-6
0	1	2	3	4	5	6	7

20	49	28	36	-12	5	40	-6
0	1	2	3	4	5	6	7

20	49	5	36	-12	28	40	-6
0	1	2	3	4	5	6	7

20	49	5	-12	36	28	40	-6
0	1	2	3	4	5	6	7

- Au delà de la case 3, chaque inversion, nous conduira au tableau du départ.

Solution Itérative de Ex 3

Solution itérative

```
Void inviterative (int T[n], int n, int i, int j )
{
    int temp;
    for (; i<j ; i++, j--)
    {
        temp = T[i];
        T[i] = T[j];
        T[j] = temp;
    }
}
```

-Le travail demandé consiste à inverser les éléments d'un tableau et le résultat doit être dans le tableau lui-même ⟶ donc nous allons obtenir un résultat, qui est le tableau modifié ⟶ Nous avons énoncé que si nous aurons un résultat alors le programme est une fonction

⟶ Hors, on a défini une procédure ⟶ Pourquoi ?

Solution Itérative de Ex 3

Fonction Vs Procédure

Fonction :

- On défini un sous-programme autant qu'une fonction, s'il y a un résultat à retourner → Dés lors la fonction appelée, est manipulée comme une variable → elle peut figurer dans une formule / équation / paramètre → Mais jamais, one peut lui affecter une valeur par une instruction d'affectation.

- Rappelez vous, le résultat retourné par la fonction, ne peut point être **une structure tableau** *(sauf s'il est manipulé comme un pointeur)*.

Solution Itérative de Ex 3

Procédure :

- Si le sous-programme, ne retourne pas un résultat, il est défini comme une procédure → Dés lors la procédure appelée est considérée comme une instruction) part entière → elle figure dans une ligne séparée et ne peut point figurée dans d'autres instructions.

Mais

- Si le résultat du sous-programme est une structure tableau → il sera défini autant que procédure et non une fonction.

Solution Itérative de Ex 3

- Alors, le tableau est passé comme paramètre.

- Il est plus pratique, de suivre le tableau par sa dimension → la dimension du tableau est passée autant que paramètre aussi.

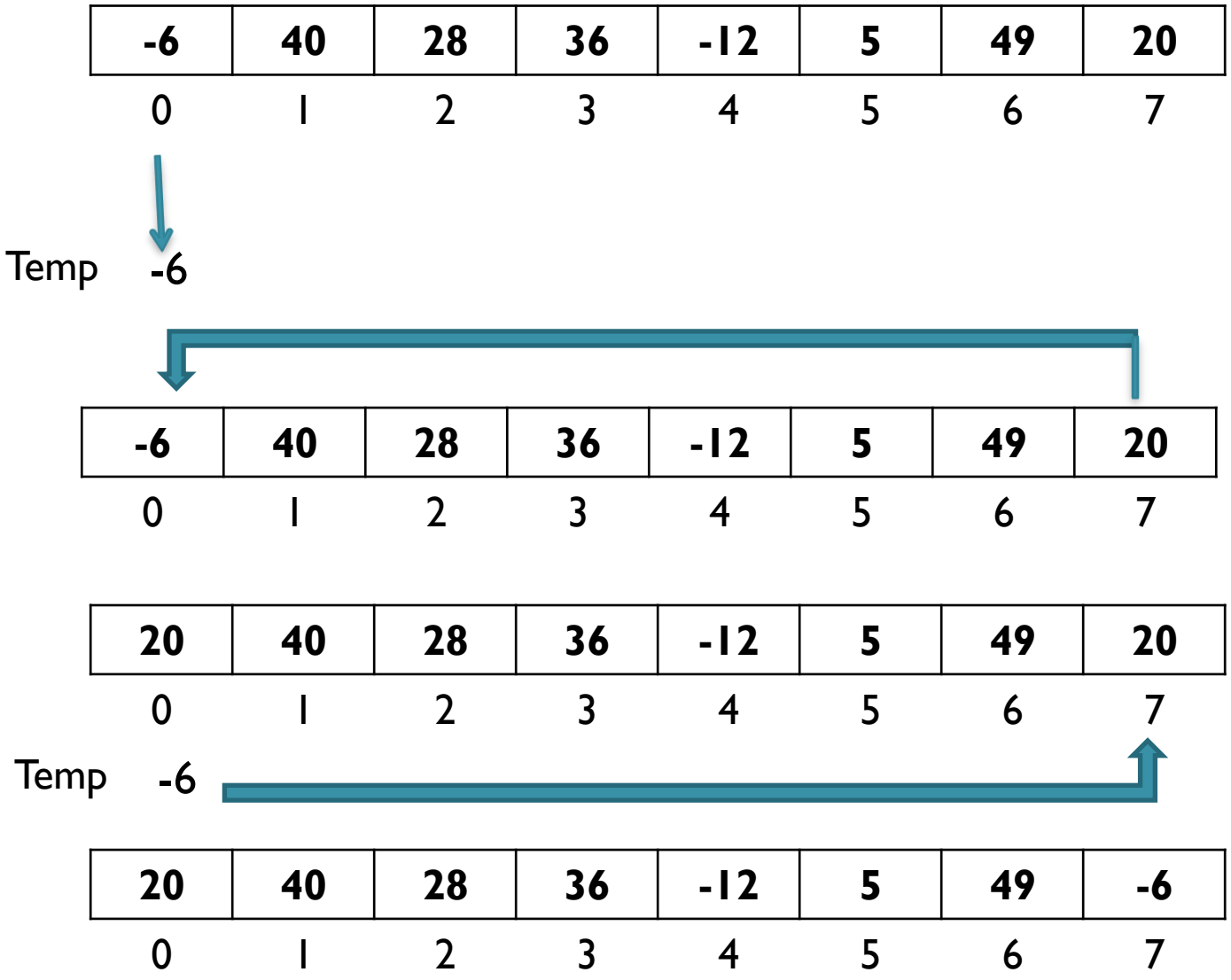
- Vous avez observé qu'il y a aussi deux autres paramètres : i , j

-i, j ; nous permettent d'indiquer à la procédure inviter où doit elle commencer.

- La variable temp est une variable locale à la procédure

- temp est utilisée pour garder la valeur de l' élément avant de la remplacer par la valeur de l' autre case

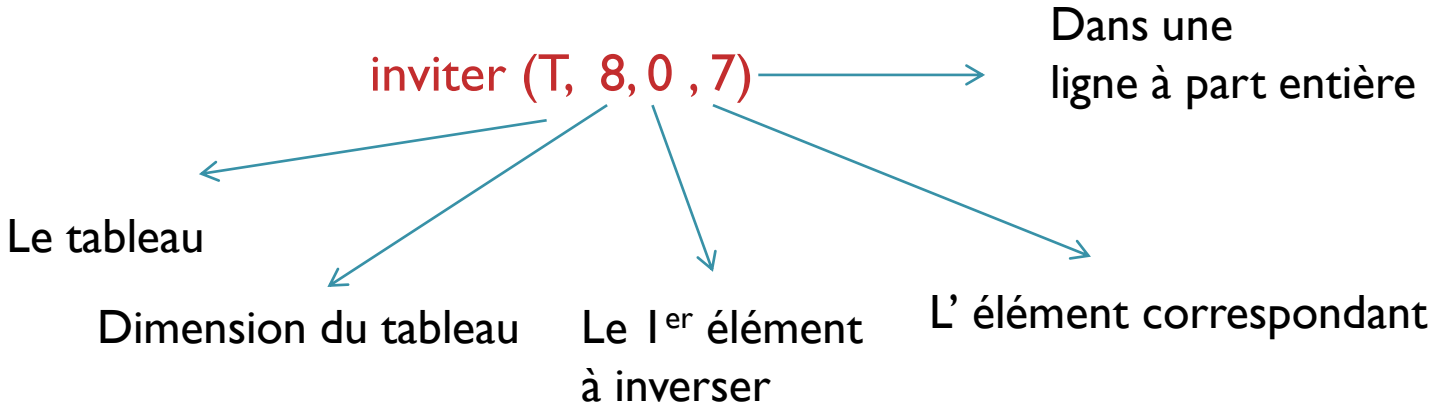
Solution Itérative de Ex 3



Solution Itérative de Ex 3

Exemple illustratif de la solution itérative :

-Nous allons prendre le même tableau et simulant l'appel de la procédure inviter .



Solution Itérative de Ex 3

- Dans la boucle For , il n y a pas la partie initialisation des compteurs ⟶

Elle utilise les valeurs passées dans les paramètres.

- La boucle For utilise deux compteurs en même temps :

  • i qui commence au début du tableau et augmente vers le milieu du tableau ⟶ donc, on incrémente i

  • J qui commence de la fin du tableau et diminue en allant au milieu du tableau ⟶ donc, on décrémente j

  • i < j est la condition qui vérifie qu'on n'est pas encore arrivé au milieu du tableau ⟶ dès qu'elle devienne fausse, on arrête .

Solution Récursive de Ex 3

Revenons à notre exemple du tableau T

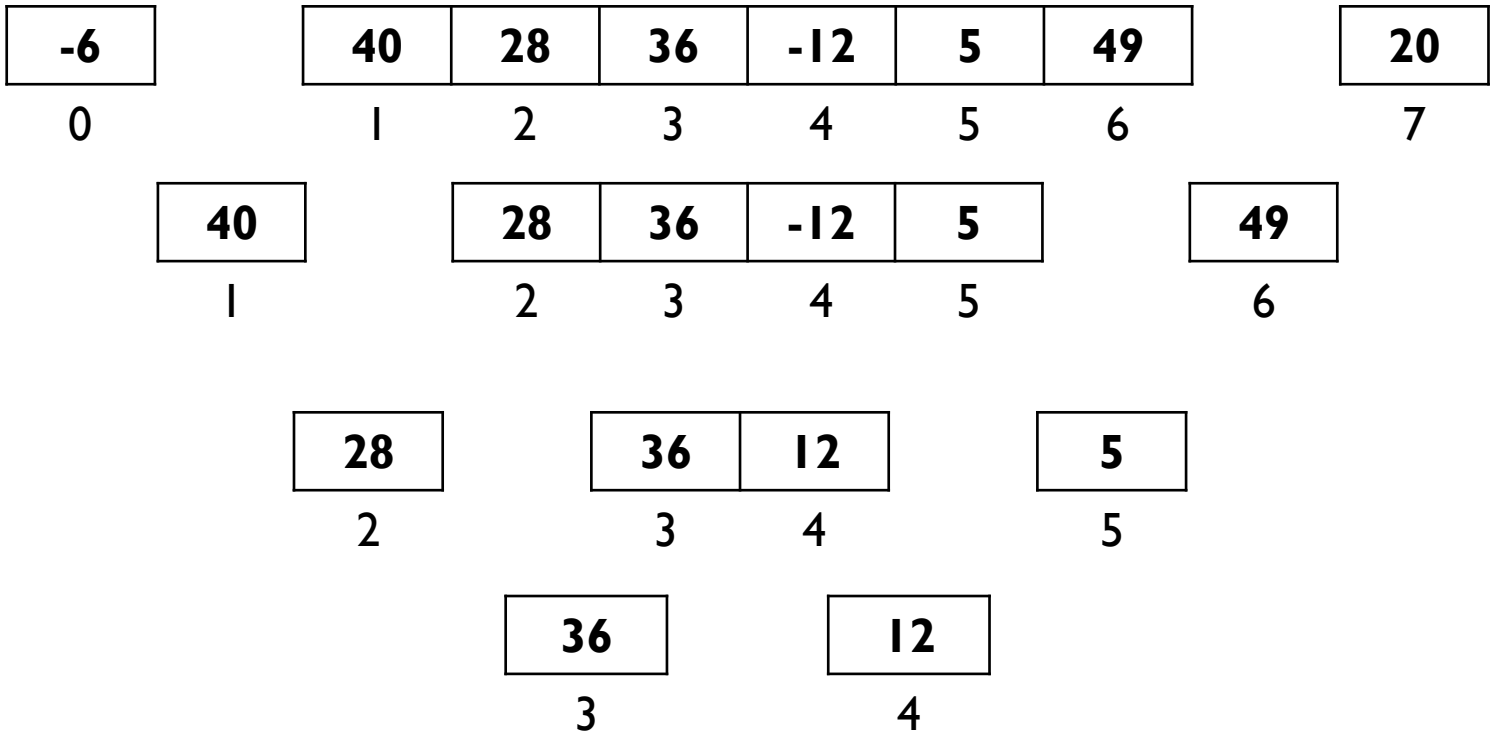
-6	40	28	36	-12	5	49	20
0	1	2	3	4	5	6	7

Définissons le d'une autre manière :

- Un tableau peut être perçu comme deux extrémités et un sous tableau
- Le sous tableau n'est autre qu'un tableau avec deux extrémités
- Et ainsi de suite jusqu'au milieu du tableau original.
- A vrai dire, cette définition n'est autre que la définition récursive d'un tableau.
- Appliquons cette nouvelle définition sur notre tableau donné en exemple

Solution Récursive de Ex 3

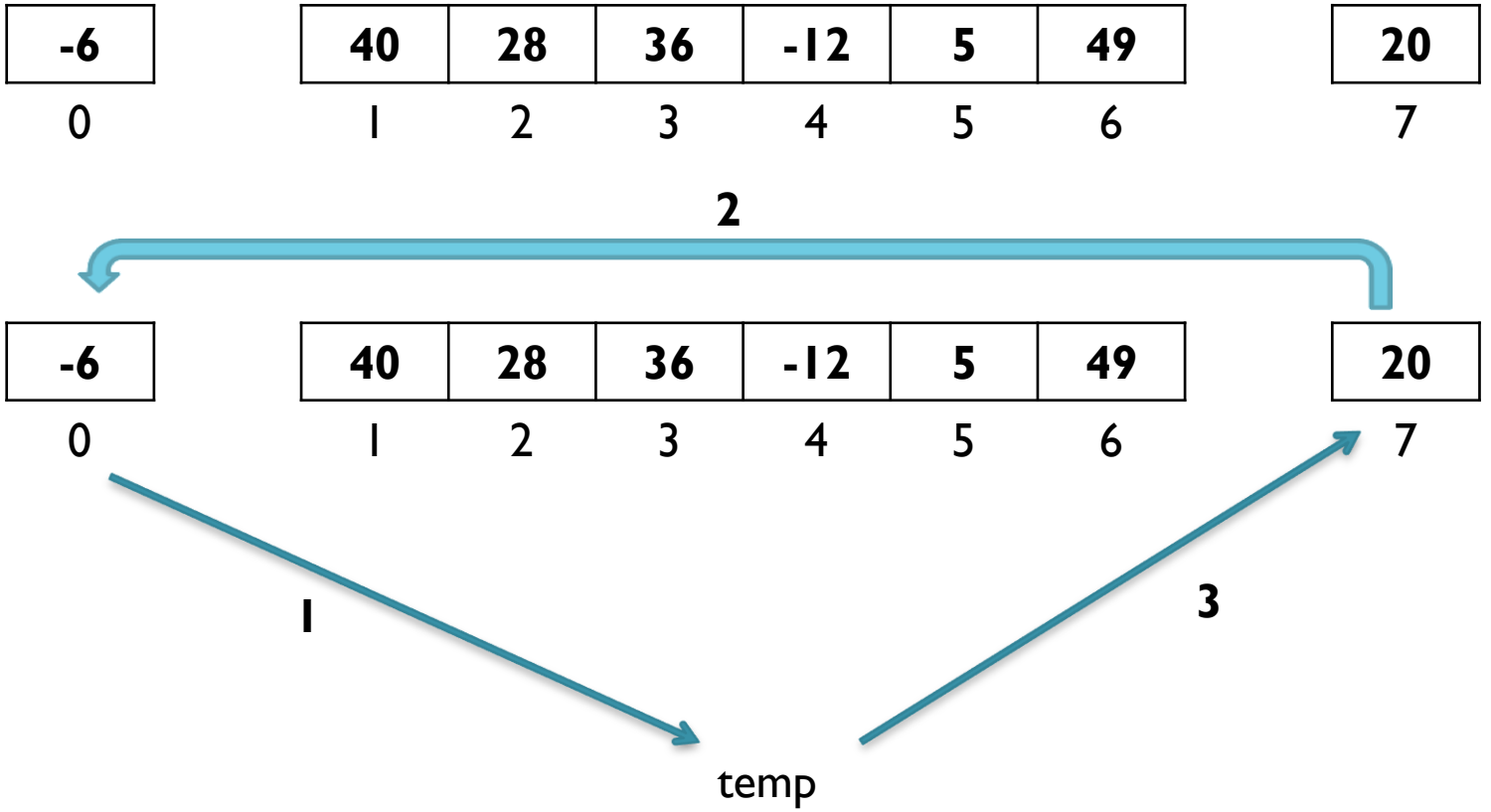
Définition récursive du tableau



Procédant à l'inversion du tableau en utilisant cette récursivité

Solution Récursive de Ex 3

Procédant à l'inversion du tableau en utilisant cette récursivité



- En réalité, on ne se préoccupe que de l'inversion des extrémités,

- Le reste est un tableau ⟶ on procédera de la même manière avec ce tableau.

Solution Récursive de Ex 3

La solution récursive demandée ressemble à sa version itérative en plusieurs points :

1- On définie une procédure ,

2- elle utilise les mêmes paramètres,

3- elle utilise un seul tableau,

4- le résultat sera conservé dans le même tableau,

5- on lui fournit les indices des extrémités autant que paramètre,

6- elle a besoin d'une variable locale *temp* pour faire l' échange,

7- la même condition y figure, pour tester l' arriver au milieu du tableau,

Mais

Elles diffèrent sur comment elles manipulent le tableau.

Solution Récursive de Ex 3

Solution récursive

```
Void invrecursive (int T[n] , int n ,  int i , int j)
{
   int temp;
   if (i < j)
   {
      temp = T[i];
     T[i] = T[j];
     T[j] = temp;
      invrecursive (T,  n , i+1 , j-1);
   }
}
```

-Dans cette solution récursive, il n y a pas de cas trivial (particulier), car dans le cas où on arrive au milieu du tableau, on arrête tout traitement.

- L' incrémentation de i et la décrémentation de j, ce fait directement dans les paramètres d'appel de la procédure .

Solution Récursive de Ex 3

Un tableau :

-6	40	28	36	-12	5	49	20
0	1	2	3	4	5	6	7

La définition récursive d'un tableau peut être :

- Une extrémité avec le reste du tableau (qui est lui-même vu comme un tableau)

-6
0

40	28	36	-12	5	49	20
1	2	3	4	5	6	7

- Ou deux extrémités avec le reste du tableau (lui-même considéré comme un tableau)

-6
0

40	28	36	-12	5	49
1	2	3	4	5	6

20
7

Fin

J'espère vous retrouvez toutes et tous en bonne santé, ainsi que votre famille.

Si vous avez des questions ou vous voulez des éclaircissements sur ce qui a été présentés, vous pouvez m' écrire à cette adresse email : hamissi_sa@yahoo.fr (le tiré du 8).

ان شاء الله نلتقي عن قريب لا فاقدين ولا مفقودين, استودعكم الله الذي لا تضيع ودائعه, السلام عليكم و رحمه الله و بركاته