

EXAMEN

Atelier de programmation II

SECTION : LGLSI 1

Nature de l'épreuve : DC <input type="checkbox"/> DS <input type="checkbox"/> EF <input checked="" type="checkbox"/>	Documents : autorisés <input type="checkbox"/> non autorisés <input checked="" type="checkbox"/>
Date de l'épreuve : 15/06/2023	Calculatrice : autorisée <input type="checkbox"/> non autorisée <input checked="" type="checkbox"/>
Durée de l'épreuve : 1h30	Session : principale <input type="checkbox"/> contrôle <input checked="" type="checkbox"/>

Exercice 1

1. Soit la déclaration suivante, `char tab[] = ""`. Que contient le tableau `tab` ?
2. Donner la déclaration d'un pointeur sur un tableau de 10 chaînes de caractères.
3. Soit la déclaration suivante : `char les[] = "ab"`. Que vaut `les[2]` ?
4. Donnez une forme équivalente à l'expression `&tab[0]`.
5. Donnez une forme équivalente à l'expression `p→champ`.
6. Dans le code suivant :

```
char ch [ ] = "Bonjour", *pc = ch ;
```

```
while (*pc)
```

```
pc++ ;
```

Que vaut `pc` après l'exécution de l'instruction `while` ? Vers quel caractère pointe-t-il ?

7. Que vaut la variable `i` à la fin de la fonction `f` ?

```
void f(int i)
```

```
{
```

```
int *j ;
```

```
j = &i ;
```

```
*j=2;
```

```
i=5;
```

```
++i;
```

```
*j=7 ;
```

```
}
```

8. Dans le code suivant :

```
int i=0, j=0 ;
```

```
j = i++ ;
```

Que vaut la variable `j` après l'instruction d'affectation ?

Exercice 2

Le but de cet exercice est d'implémenter une liste linéaire.

Dans une liste linéaire, l'insertion et la suppression d'élément peut se faire n'importe où.

Une liste linéaire peut être représentée par une structure de données composée de (Fig 1) :

- Un pointeur sur le premier élément de la liste ;
- Un pointeur sur le dernier élément de la liste ;
- Un pointeur "courant" qui est un pointeur permettant de se déplacer dans la liste.

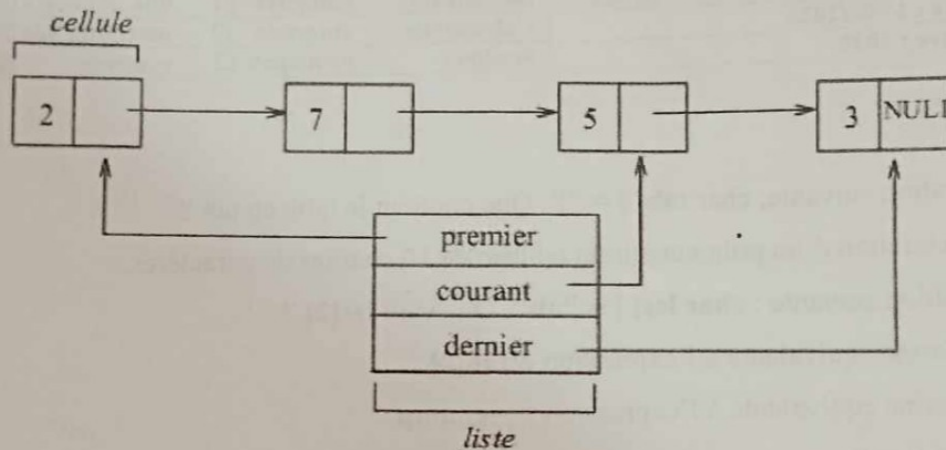


FIG. 1 - Un exemple de liste

On considère dans la suite de l'exercice que l'on implémente une liste d'entiers.

On considère dans la suite de l'exercice qu'une liste vide est une structure dans laquelle les pointeurs premier, courant et dernier sont nuls.

1. Décrivez les structures de données **liste** et **cellule** en langage C.
2. Ecrivez le code C de la fonction **Init** qui initialise une liste vide.
3. Ecrivez le code C de la fonction **Inserer** qui insère un élément dans la liste juste avant le pointeur courant ; le pointeur courant pointe ensuite sur ce nouvel élément. Vous considérerez trois cas : 1) la liste est vide, 2) le pointeur courant est sur le premier élément et 3) le cas général.
4. Ecrivez le code de la fonction **Backup** qui parcourt la liste et affiche la valeur de tous les éléments de la liste, séparés par des espaces.
5. Ecrivez une fonction **main** permettant de remplir une liste de N entiers et d'afficher les éléments de la liste.

Bon courage