Travaux dirigés n°9 (2h15): Procédures et fonctions(2): Pointeurs/paramètres

---

Objectifs

- maîtriser la notion de pointeur et la gestion de la mémoire lors des passages de paramètres.
- écrire des fonctions/procédures

Exercice 1:

a) Écrire une fonction estPremier qui retourne *TRUE* si un entier (≥ 2) est un nombre premier, *FALSE* sinon. Vous utiliserez la fonction divise de l'exercice 1 du TD précédent.
Rappel : un nombre entier est premier si ses 2 seuls diviseurs sont 1 et lui-même.

b) Écrire un algorithme principal qui teste si un entier est premier et affiche "n est premier" ou "n n'est pas premier" suivant les cas.

Exercice 2: Pointeurs.

"Dessiner la mémoire" du programme suivant et indiquer les affichages qui ont lieu

```
int main()
{
 int n=5, m;
 float s, r=0.3 ;
 int* pp ;
 float* pr ;

 pp=&m ;
 *pp=n ;
 printf(“ %d -”, m) ;
 pr=&r ;
 s=*pr ;
 printf(“ %f -”, s) ;
}
```

Exercice 3: Passage de paramètres par adresse.

On reprend la procédure C de décalage circulaire vers la gauche d’un tableau de réels vu au dernier TD.

```
void decaleCircGauche(int t[],int nb)
{
        int i ;
        float aux ;

        aux=t[0] ;
        for (i=0 ; i<nb –1 ; i++)
                t[i] = t[i+1] ;
        t[nb-1]=aux ;
}
```

On dispose aussi de la procédure :

```
procedure afficheTableau( →int t[], →int nb);
```

Soit le programme principal suivant :

```
int main()
{
        int tab[6] = {4, 8, 6};
        int nbEff = 3;
        decaleCircGauche(tab, nbEff);
        afficheTableau( tab, nbEff);
}
```

Dérouler le programme et "dessiner la mémoire" très précisément au fur et à mesure de l'exécution du programme.

Exercice 4: Paramètres et pointeurs

1. Soit la procédure suivante :

```
void faitQuoi( int x, int y, int d)
{
 x = x + d;
 y = y+ d;
}
```

Soit le programme principal suivant :

```
int main()
{
 int a = 8;
 int b = 2;
 int c = 4;
```

```
 faitQuoi (a, b, c);
 printf("a=%d, b=%d", a, b);
 return(1);
}
```

Dérouler le programme et "dessiner la mémoire" très précisément au fur et à mesure de l'exécution du programme. Indiquer l'affichage qui aura lieu.

2. Même question avec la procédure et l'appelant suivants

```
void faitQuoiBis( int *px, int *py, int d)
{
 *px = *px + d;
 *py = *py+ d;
}
```

Soit le programme principal suivant :

```
int main(void)
{
 int a = 8;
 int b = 2;
 int c = 4;
 faitQuoiBis (&a, &b, c);
 printf("a=%d, b=%d", a, b);
 return(1);
}
```

Exercice 5:

a) Écrire une fonction geometrique(…) qui retourne la valeur $1 + r + r^2 + \ldots + \ldots$ (r réel vérifiant $0 < r < 1$) sachant qu'on n'intégrera dans la somme que les termes strictement supérieurs à un certain seuil fourni (*seuil* réel vérifiant $0 < seuil < r$).
b) Écrire un algorithme principal testant la fonction.