

## Programmation en C

### TP N°3

#### Exercice 1 :

Ecrire une fonction paramétrée qui retourne l'inverse d'un nombre entier donné.(1234=>4321)

#### Exercice 2 :

Ecrire une fonction permettant de calculer le PGDC de deux nombres entiers positifs A et B(cours).

#### Exercice 3 :

Ecrire une fonction qui permet de calculer le factoriel d'un nombre donné( voir cours).

#### Exercice 4 :

Ecrire une fonction qui permet de calculer la puissance d'un entier donné.

#### Exercice 5 : Travail à maison

Ecrire une fonction qui calcule la valeur de la fonction d'Ackermann « A » définie pour  $m > 0$  et  $n > 0$  par :

$$\begin{aligned} A(m, n) &= A(m-1, A(m, n-1)) \text{ pour } n > 0, m > 0 ; \\ A(0, n) &= n+1 \text{ pour } n > 0 ; \\ A(m, 0) &= A(m-1, 1) \text{ pour } m > 0 ; \end{aligned}$$

#### Exercice 6 :

Ecrire une fonction qui fournit le nombre de chiffres d'un entier donné.

#### Exercice 7 :

Ecrire une procédure qui permet de dessiner la lettre X, à l'aide d'espaces et d'une "lettre" fournie par l'utilisateur, auquel on demande aussi la "hauteur" du dessin qu'il désire obtenir. Par exemple : avec les réponses a et 5, elle donnera :



#### Exercice 8 :

Ecrire une procédure paramétrée, qui permet l'échange des contenus de deux paramètres formels par valeur A et B. Appeler cette procédure dans un programme principal. On écrira les résultats dans le corps de la procédure, et dans le programme principal.

#### Exercice 9 :

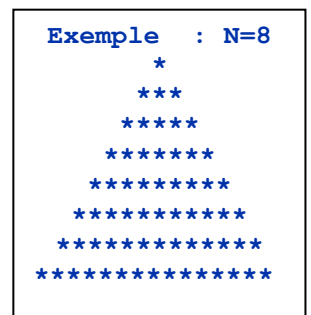
Répéter l'exercice précédent mais avec un passage par adresse des paramètres A et B.

#### Exercice 10 :

Ecrire une procédure qui affiche tous les nombres premiers qui sont compris entre 1 et 200.

#### Exercice 11 :

Affichez un triangle isocèle formé d'étoiles sur N lignes (N est fourni au clavier).



**Exercice 12 : Travail à maison**

Calculer pour une valeur X donnée du type réel la valeur numérique d'un polynôme de degré n:

$$P(X) = A_n X^n + A_{n-1} X^{n-1} + \dots + A_1 X + A_0$$

Les valeurs de n, des coefficients  $A_n, \dots, A_0$  et de X seront entrées au clavier.

Utiliser le schéma de Horner qui évite les opérations d'exponentiation lors du calcul:

$$\begin{array}{c} A_n \\ \underbrace{\quad} \\ * X + A_{n-1} \\ \underbrace{\quad} \\ * X + A_{n-2} \\ \underbrace{\quad} \\ \dots \\ * X + A_0 \end{array}$$

**Exercice 13 : Travail à maison**

Ecrire une procédure qui affiche la table des produits pour N variant de 1 à 10 :

X*Y	I	0	1	2	3	4	5	6	7	8	9	10
0	I	0	0	0	0	0	0	0	0	0	0	0
1	I	0	1	2	3	4	5	6	7	8	9	10
2	I	0	2	4	6	8	10	12	14	16	18	20
3	I	0	3	6	9	12	15	18	21	24	27	30
4	I	0	4	8	12	16	20	24	28	32	36	40
5	I	0	5	10	15	20	25	30	35	40	45	50
6	I	0	6	12	18	24	30	36	42	48	54	60
7	I	0	7	14	21	28	35	42	49	56	63	70
8	I	0	8	16	24	32	40	48	56	64	72	80
9	I	0	9	18	27	36	45	54	63	72	81	90
10	I	0	10	20	30	40	50	60	70	80	90	100

## Programmation en C

### TP N°4

#### Exercice 1      TD

Soit les déclarations suivantes :

**int A = 1, B = 2, C = 3 ;**

**int \*P1, \*P2;**

Complétez le tableau suivant pour chaque instruction.

	A	B	C	P1	P2
<i>init.</i>	1	2	3	/	/
<b>P1=&amp;A</b>	1	2	3	&A	/
<b>P2=&amp;C</b>	1	2	3	&A	&C
<b>*P1=(*P2)++    A=C++</b>	3	2	4	&A	&C
<b>P1=P2</b>	3	2	4	&C	&C
<b>P2=&amp;B</b>	3	2	4	&C	&B
<b>*P1-=*P2</b>	3	2	2	&C	&B
<b>++*P2    ++B</b>	3	3	2	&C	&B
<b>*P1*=*P2    C*=B</b>	3	3	6	&C	&B
<b>A= *P1++; A=*P1; P1++</b>	6	3	6		

#### Exercice 2      TD

Soit **P** un pointeur qui pointe sur un tableau **A** :

**int A[] = {11, 13, 7, 44, 66, 77, 88, 99, 100};**

**int \*P;**

**P = A;**

Quelles valeurs ou adresses fournissent ces expressions:

- a) **\*P+2**                      11+2=13                      b) **\*(P+2)** =7                      c) **&P+1**                      adresse derrier adresse  
d) **&A[4]-3**                      p+4-3=p+1 =&A[1]  
e) **A+3**                      =&A[3]  
f) **&A[7]-P**                      =                      **P+7-P=7**  
g) **A[0]=12; P+(\*P-10)**                      = P+(12-10)=P+2 =&A[2]  
h) **\*(P+(P+8)-A[7])**                      = **\*(P+A[8]-99)**                      = **\*(P+90-99)**                      = **\*(P-9)**

#### Exercice 3      TD

Quels résultats fournira ce programme :

```
#include <stdio.h>
main() {
    int t [3]; int i, j; int *adt;
    for (i=0, j=0; i<3; i++) t[i] = j++ + i;                      /* 1 */    0 2 4
    for (i=0; i<3; i++) printf("%d \n ", t[i]);                      /* 2 */
    for (i=0; i<3; i++) printf("%d \n ", *(t+i));                      /* 3 */
    for (adt = t; adt < t+3; adt++) printf("%d \n ", *adt) ;                      /* 4 */
    for (adt = t+2; adt>=t; adt--) printf("%d \n ", *adt);                      /* 5 */
}
```

#### Exercice 4

Ecrire un programme qui permet de saisir un entier N représentant le nombre des notes à saisir, saisit ces notes dans un tableau et calcule la somme, la moyenne, la valeur maximale et la valeur minimale.

#### Exercice 5

Écrire une fonction qui ne renvoie aucune valeur et qui détermine la valeur maximale et la valeur minimale d'un tableau d'entiers (à un indice) de taille quelconque. Il faudra donc prévoir 4 arguments : le tableau, sa dimension, le maximum et le minimum. Écrire un petit programme d'essai.

**void maxmin (int t[], int n, int \*admax, int \*admin)**

#### Exercice 6      TD

Considérons l'exécution des instructions suivantes :