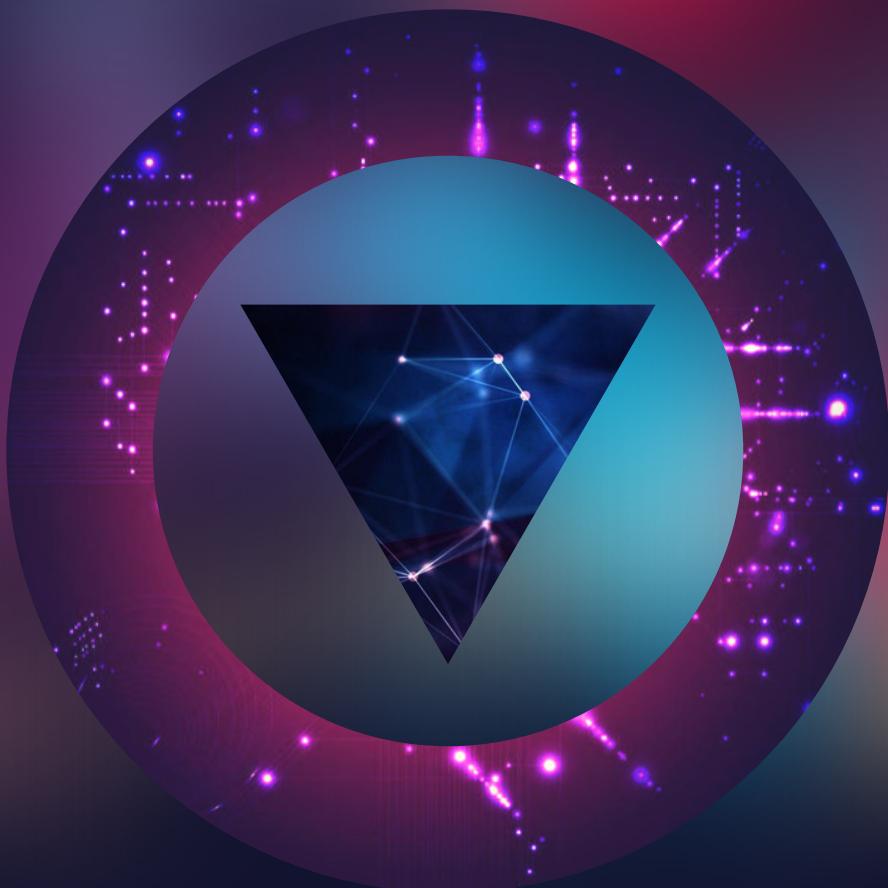


PROGRAMMATION 1

Contrôles corrigés

FACULTE: FES



S M I A
S T U D I E S

Examen du demi module informatique 2
Section de rattrapage
Durée : 1 h 30 mn

Pour chaque question, reporter sur votre feuille le numéro de la question et votre réponse.

Exercice I : Questions de cours (5 pts)

1. La compilation d'un programme C permet de créer un module objet ?
 A. Oui
 B. Non
2. Dans les étapes de la réalisation d'un programme en langage C, l'éditeur de liens permet de :
 A. chercher les erreurs syntaxiques dans un programme C
 B. rechercher dans la bibliothèque standard les modules objets nécessaires
 C. faire une réservation mémoire pour les variables
3. Soit l'algorithme suivant :

Variabes i, j : entiers
k : booléen
Début
 i ← 3, j ← 7; j ← i;
 Si ($i < j$) alors
 k ← Vrai
 Sinon
 k ← Faux
 FinSi
 écrire(j, ", ", k)
Fin

L'exécution de ce programme affichera

- A. 3, Vrai
- B. 3, Faux
- C. 7, Vrai
- D. 7, Faux

4. Soit le programme suivant :

```
int T[8];  
int i;  
for (i = 0 ; i < 8 ; i = i+1)  
    T[i] = i + 1;  
printf("%d", T[3]);
```

L'exécution de ce programme affichera

- A. 1
- B. 2
- C. 3
- D. 4

5. Soit le programme suivant :

```
int fonction1 (int i) { return (i % 2); }  
int fonction2 (int i) { return (++i); }  
int main () {  
    int a, b;  
    a = fonction1 (5);  
    b = fonction2 (1);  
    printf("a=%d, b=%d\n", a, b)  
}
```

X Vrai

L'exécution de ce programme affichera :

- A. $a = 2, b = 1$
- B. $a = 2, b = 2$
- C. $a = 1, b = 1$
- D. $a = 1, b = 2$

Exercice II : Algorithmique (3 pts)

Ecrire un algorithme qui demande à l'utilisateur de saisir un entier n positif et calcule le factoriel de n .

Exercice III : Fonction en langage C (8 pts)

Le but de cet exercice est de réaliser une fonction *Calcul*, en langage C, permettant d'effectuer les opérations de base d'une calculatrice. La fonction *Calcul* comprend trois variables passées en paramètre : deux entiers et un caractère qui représente l'opération à effectuer. La fonction retourne le résultat de l'opération demandée.

1. Ecrire la fonction *Calcul* qui doit gérer les opérateurs : + (la somme), - (la soustraction), * (la multiplication) et / (la division).

NB : pensez à utiliser l'instruction *switch ... case*

2. Modifier votre fonction pour qu'elle calcule la puissance. Par exemple si je saisis les deux entiers suivants 3, 4 et je choisis l'opération p , la fonction retournera 81 (équivalent à 3^4)

Attention : il faut tester si l'exposant est un entier positif, nul ou négatif.

Exercice IV : Tableau en C (4 pts)

Le but de cet exercice est de réaliser un programme permettant de permuter les éléments d'un tableau :

Par exemple, si on considère le tableau suivant :

350	56	78	20	2	13	180
-----	----	----	----	---	----	-----

Voici l'ordre des éléments du tableau après permutation.

180	13	2	20	78	56	350
-----	----	---	----	----	----	-----

On a échanger $T[i]$ avec $T[n-i-1]$ où n est la taille du tableau.

Le programme doit réaliser les fonctions suivantes :

1. saisir la taille d'un tableau (maximum 30 éléments)
2. remplir le tableau par des entiers
3. permuter les éléments de ce tableau
4. afficher le tableau après permutation

Correction (examen du rattrapage)

Exercice I : Questions de cours (5 pts)

1. A (1)
2. B (1)
3. C (1)
4. D (1)
5. D (1)

Exercice II : Algorithmique (3 pts)

Algorithme Factoriel

Variables fact, n, i : entiers

Début

Répéte

 écrire("entrer un entier positif")
 lire(n)
 jusqu'à ($n > 0$)

 fact = 1

 pour i allant de 1 à n faire
 fact = fact * i

 fin pour
 écrire (" le factoriel de ", n "est" fact)
Fin

Exercice III : Fonction en C (8 pts)

1. (4.5 pts)

float (ou int) Calcul(int a, int b, char c){
int (ou float) R;

switch(c){
 case '+': R = a + b; break;
 case '-': R = a - b; break;
 case '*': R = a * b; break;
 case '/': R = a / b; break;
 default : break;
}
return(R);

2. (3.5 pts)

float Calcul(int a, int b, char c){
 int i;
 float puiss, R;

switch(c){

 case 'p':

 if(b == 0) return(1);

 else {
 puiss=1;

 }

 for(i = 0; i < abs(b); i++)
 puiss = puiss * a;

 if(b > 0)

 return(puiss);

 else
 return(float(1/puiss));

 } break;

 default : break;

Exercice IV : Tableau en C (4 pts)

```
#include <stdlib.h>
#include <stdio.h>
int main()
{
    int n, i, T[30], tmp;
    //saisie la taille
    do{
        scanf("%d", &n);
        while(n < 1 || n > 30);
    } (0.5)
    //saisie des notes
    for (i = 0; i < n/2 ; i++)
        scanf("%d", &T[i]);
    //permutation
    for (i = 0; i < n/2; i++){
        tmp = T[i];
        T[i] = T[n - i - 1];
        T[n - i - 1] = tmp;
    }
    //affichage
    for (i = 0; i < n; i++)
        printf("%d", T[i]);
}
return EXIT_SUCCESS;
```

Examen du demi module : Informatique 2
Durée : 1 h 30 mn

Exercice I : (5 pts)

Cet exercice est composé d'un ensemble de questions réparties en deux colonnes. Vous n'allez pas répondre à toutes les questions, mais seulement aux questions de votre choix. Par exemple, vous pouvez choisir de répondre aux questions du choix A. Dans ce cas, vous ne devez pas répondre aux questions du choix B. Rapporter dans votre feuille de rédaction, votre choix et le numéro de la question suivie de la réponse.

1. gec est :

- a. Un identificateur du langage C.
- b. Une commande qu'on tape dans la fenêtre de commande
- c. Un mot-clef du langage C
- d. Un opérateur du langage C

2. while est :

- a. Un identificateur du langage C
- b. Une commande qu'on tape dans la fenêtre de commande
- c. Un mot-clef du langage C
- d. Un opérateur du langage C

3. Dans le programme main() {int a ; a=3%5 ;}, % est :

- a. Un identificateur du langage C
- b. Une commande qu'on tape dans la fenêtre de commande
- c. Un mot-clef du langage C
- d. Un opérateur du langage C

5. = permet en langage C de :

- a. réaliser une affectation
- b. tester une égalité
- c. comparer deux identificateurs
- d. convertir un int en float

Exercice II : Algorithmique (4pts)

Ecrire un algorithme qui calcule le produit scalaire de deux vecteurs d'entiers U et V (de même dimension).

Exercice III : Fonction en langage C (5 pts)

1. Ecrire une fonction MAX en langage C qui à partir de deux entiers donnés, retourne le maximum.
2. Ecrire une fonction MAX3 en langage C qui retourne le maximum de trois entiers en faisant appel à la fonction MAX2.

Exercice IV : Tableau en C (6 pts)

Ecrire un programme en langage C qui détermine la plus grande et la plus petite valeur dans un tableau d'entiers A. Afficher ensuite la valeur et la position du maximum et du minimum. Si le tableau contient plusieurs maxima ou minima, le programme retiendra la position du premier maximum ou minimum rencontré.

1-Soit P un pointeur qui 'pointe' sur un tableau A:

```
int A[] = {14, 25, 36, 47, 58, 69, 80, 91, 92};  

int *P;  

P = A;
```

Quelles valeurs ou adresses fournissent ces expressions

- a) *(P+2) → 36
- b) &P+2 →
- c) &A[7]-P →
- d) *(P+*(P+8)-A[7]) →

2-Soit le programme suivant :

```
int main() {  

    int A=2, B=3, C=4,  

    int *P1, *P2;  

    P1=&A; P2=&C;
```

Quelles valeurs fournissent ces expressions

- a) *P1=(*P2)+1;
- b) P2=&B;
- c) *P1=*P2;
- d) *P2=*P1/*P2;

Correction d'examen S4 SVI/STU
Pr H. Tairi

Exercice 1 : (5 pts)

1-	<code>int A[] = {14, 25, 36, 47, 58, 69, 80, 91, 92};</code>		
a)	<code>*(&P+2)</code>	36	0,5pt
b)	<code>&P+2</code>		0,5pt
c)	<code>&A[7]-P</code>	A[2]	0,5pt
d)	<code>*(P+*(P+8)-A[7])</code>	indice 7	0,5pt
2-		25	0,5pt

	A	B	C	P1	P2	
int	2	3	4			
P1=&A	2	3	4	P1=&A		
P1=&C	2	3	4	&A	&C	
*P1=(*P2)++	4	3	5	&A	&C	0,5pt
P2=&B	4	3	5	&A	&B	0,5pt
*P1 *= *P2	12	3	5	&A	&B	1pt

Exercice II : Algorithmique (4pts)

Algorithmes de calcul

Variable tableau U[50];

tableau $U[50]$: entier

N, l, PS ; entier

Début

```
    ecrire("Dimension des tableaux (max.50) : ");  
    lire(N);
```

pour i allant de 0 à $N-1$ (0,5pt)

écrire("Element : ", x)

fin $\text{line}(\cup[I])$

ANSWER

our Fallant de 0 à

pour l'allant de 0 à N-1
début

```
    ecrire("Elément : ", I)
    lire(V[i])
```

fin lire (V[I])

our Jallant, A., a man of 37

our i allant de 0 à $N-1$

$PS \leftarrow PS + U[I] * V[I];$

crire ("Produit scalaire : ", p)

(Produit scalaire : ", PS);

Exercice III : Fonction en langage C (5 pts)

1)

```
Int max2 (int x, int y) {  
    If (x<y)  
        Return(y);  
    Else  
        Return(x);  
}
```

(0,5pt)

(1,5pt)

2) Int max3 (int x, int y, , int z) {
 If (x<y)
 Return(max2(y,z));
 Else
 Return(max2(x,z));

(0,5pt)

(2,5pt)

6

Exercice IV : Tableau en C (6 pts)

```
#include <stdio.h>  
main()  
{  
    /* Déclarations */  
    int A[50]; /* tableau donné */  
    int N; /* dimension */  
    int I; /* indice courant */  
    int MIN; /* position du minimum */  
    int MAX; /* position du maximum */  
    /* Saisie des données */  
    printf("Dimension du tableau (max.50) : ");  
    scanf("%d", &N);  
    for (I=0; I<N; I++)  
    {  
        printf("Elément %d : ", I);  
        scanf("%d", &A[I]);  
    }  
    /* Affichage du tableau */  
    printf("Tableau donné :\n");  
    for (I=0; I<N; I++)  
        printf("%d ", A[I]);  
    printf("\n");  
    /* Recherche du maximum et du minimum */  
    MIN=0;  
    MAX=0;  
    for (I=0; I<N; I++)  
    {  
        if(A[I]>A[MAX]) MAX=I;  
        if(A[I]<A[MIN]) MIN=I;  
    }  
    /* Edition du résultat */  
    printf("Position du minimum : %d\n", MIN);  
    printf("Position du maximum : %d\n", MAX);  
    printf("Valeur du minimum : %d\n", A[MIN]);  
    printf("Valeur du maximum : %d\n", A[MAX]);  
    return 0;}
```

(0,5pt)

(0 pt)

(0,5pt)

(1pt)

(1pt)

(1pt)

Examen du demi module informatique 2
Section de rattrapage
Durée : 1 h 30 mn

Exercice I : (5 pts)

Pour chaque question, trouvez la valeur des variables qui figurent en commentaire

Question 1

```
void main(void)
{
    int x = 10;
    int y, z;
    x *= y = z = 4;           // x = 40
```

Question 3

```
void main(void)
{
    int x = 3, y = 2;
    int a;
    a = x < y ? x++ : y++; // x = 3 y = 3 a = 2
```

Question 2

```
void main(void)
{
    int x = 2, y = 1, z = 0;
    z += -x++ + ++y;        // x = 3 y = 2 z = 0
```

Question 4

```
void main(void)
{
    int x = 3, z = 0;
    z = x / ++x;            // attention au piège!
                           // x = 2 z = 0
```

Exercice II : Algorithmique (4,5 pts)

Ecrire avec la boucle POUR un algorithme qui permet de faire la somme d'une suite de nombre entrée par l'utilisateur. Faire la même chose en comptant par pas de -1.

Exercice III : Fonction en langage C (4 pts)

Ecrire un programme en langage C se servant d'une fonction F pour afficher la table de valeurs de la fonction définie par :

$$F(x) = \sin(x) + \cos(x) + \ln(x) - \sqrt{x}$$

où x est un entier compris entre 1 et 20.

Exercice IV : Tableau en C (6,5 pts)

Ecrire un programme en langage C qui lit la dimension N d'un tableau T du type int (dimension maximale: 30 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Copiez ensuite toutes les composantes strictement positives dans un deuxième tableau TPOS et toutes les valeurs strictement négatives dans un troisième tableau TNNEG.
Afficher les tableaux TPOS et TNNEG.

Correction (examen du rattrapage S4 SVI/STU)

Exercice 1

Question 1 $x = 40$

Question 2 $x = 3 \quad y = 2 \quad z = 0$

1pt

Question 3 $x = 3 \quad y = 3 \quad a = 2$

1.5pt

Question 4 $x = 4 \quad z = \text{indéfini} (0 \text{ ou } 1)$

1.5pt

1pt

Exercice 2

Algorithme : somme_nombre

Var : k, nb, n, somme : entier

Début :

 Somme <= 0

0.25pt

 Ecrire (« combien voulez-vous entrer de nombres ? »)

0.5pt

 Lire (nb)

0.25pt

 Pour k de 1 à nb faire

 Début

0.25pt

 Lire (n)

1pt

 Somme <= somme + n

0.25pt

 Fin

0.25pt

 Ecrire (somme)

0.25pt

Fin

Même programme par pas de -1 :

Algorithme : somme_entier

Var : k, nb, n, somme : entiers

Début :

 Somme <= 0

0.25pt

 Ecrire (« combien voulez-vous entrer de nombres ? »)

0.5pt

 Lire (nb)

0.25pt

 Pour k de nb à 1 par pas de -1 faire

 Début

0.25pt

 Lire (n)

1pt

 Somme <= somme + n

0.25pt

 Fin

0.25pt

 Ecrire (somme)

0.25pt

Fin

Exercice 3

```
#include <stdio.h>
#include <math.h>
main()
{
    /* Prototypes des fonctions appelées */
    double F(int X);
    /* Variables locales */
    int I;
    /* Traitements */
    printf("\nX\tF(X)\n");
    for (I=1 ; I<=20 ; I++)
        printf("\t%d\t%lf\n", I, F(I));
    return 0;
}
double F(int X)
{
    return sin(X)+log(X)-sqrt(X);
}
```

0.5pt

0.5pt

0.5pt

1pt

1pt

Exercice 4

```
#include <stdio.h>
main()
{
    /* Déclarations */
    /* Les tableaux et leurs dimensions */
    int T[50], TPOS[50], TNNEG[50];
    int N, NPOS, NNEG;
    int I; /* indice courant */

    /* Saisie des données */
    printf("Dimension du tableau (max.50) : ");
    scanf("%d", &N);
    for (I=0; I<N; I++)
    {
        printf("Elément %d : ", I);
        scanf("%d", &T[I]);
    }
    /* Affichage du tableau */
    printf("Tableau donné :\n");
    for (I=0; I<N; I++)
        printf("%d ", T[I]);
    printf("\n");

    /* Initialisation des dimensions de TPOS et TNNEG */
    NPOS=0;
    NNEG=0;
    /* Transfer des données */
    for (I=0; I<N; I++)
    {
        if (T[I]>0)
            TPOS[NPOS]=T[I];
        NPOS++;
    }
    if (T[I]<0)
        TNNEG[NNEG]=T[I];
    NNEG++;

    /* Edition du résultat */
    printf("Tableau TPOS :\n");
    for (I=0; I<NPOS; I++)
        printf("%d\n", TPOS[I]);
    printf("\n");
    printf("Tableau TNNEG :\n");
    for (I=0; I<NNEG; I++)
        printf("%d\n", TNNEG[I]);
    printf("\n");
    return 0;
}
```

1pt

0.5pt

1pt

0.5pt

0.5pt

1pt

1pt

0.5pt

0.5pt

Exercice 1

(TD) N° 5: Langage C

Ecrire un programme qui calcule x^n , où x est un nombre réel de type double et n un entier, tous deux entrés au clavier. On écrira le programme en utilisant une boucle for, puis une boucle while.

Exercice 2 Produit scalaire de deux vecteurs

Ecrire un programme qui calcule le produit scalaire de deux vecteurs d'entiers U et V (de même dimension).

Exercice 3 Maximum et minimum des valeurs d'un tableau

Ecrire un programme qui détermine la plus grande et la plus petite valeur dans un tableau d'entiers A. Afficher ensuite la valeur et la position du maximum et du minimum. Si le tableau contient plusieurs maxima ou minima, le programme retiendra la position du premier maximum ou minimum rencontré.

Exercice 4

Ecrire un programme se servant d'une fonction F pour afficher la table de valeurs de la fonction définie par

$$f(x) = \sin(x) + \ln(x) - \sqrt{x}$$

où x est un entier compris entre 1 et 10.

Exercice 5

Ecrire une fonction MIN et une fonction MAX qui déterminent le minimum et le maximum de deux nombres réels.

Ecrire un programme se servant des fonctions MIN et MAX pour déterminer le minimum et le maximum de quatre nombres réels entrés au clavier

Exercice 6

Rédiger un algorithme de recherche simultanée du minimum et du maximum d'un ensemble de N éléments

Exercice 7 autrement c'est l'exercice 1

Concevoir la fonction puissance capable de calculer les puissances entières positives de nombres réels et écrire un programme de test qui l'utilise.

Exercice 1

```
/***
 *      calcule de x^n
 */
#include <stdio.h>
main()
{
    int n, i;
    double x, puissance;
    printf("Entrez x : ");
    scanf("%lf", &x);
    printf("Entrez n : ");
    scanf("%d", &n);
    printf("\n Calcul de x^n avec une boucle for\n");
    for (i = 1, puissance = 1; i <= n; i++)
        puissance *= x;
    printf("(%g) ^ %d = %g\n", x, n, puissance);

    printf("\n Calcul de x^n avec une boucle while\n");
    i = 0;
    puissance = 1;
    while (i < n)
    {
        puissance *= x;
        i++;
    }
    printf("(%g) ^ %d = %g\n", x, n, puissance);
}
```

Solutions**Exercice 2**

```
#include <stdio.h>
main()
{
    /* Déclarations */
    int U[50], V[50]; /* tableaux donnés */
    int N; /* dimension */
    int I; /* indice courant */
    long PS; /* produit scalaire */
    /* Saisie des données */
    printf("Dimension des tableaux (max.50) : ");
    scanf("%d", &N);
    printf("Premier tableau **\n");
    for (I=0; I<N; I++)
    {
        printf("Elément %d : ", I);
        scanf("%d", &U[I]);
    }
    printf("Deuxième tableau **\n");
    for (I=0; I<N; I++)
    {
        printf("Elément %d : ", I);
        scanf("%d", &V[I]);
    }
    /* Calcul du produit scalaire */
    for (PS=0, I=0; I<N; I++)
        PS += (long)U[I]*V[I];
}
```

```
    /* Edition du résultat */
    printf("Produit scalaire : %d\n", PS);
    return 0;
}
```

Exercice 3

```
#include <stdio.h>
main()
{
    /* Déclarations */
    int A[50]; /* tableau donné */
    int N;      /* dimension */
    int I;      /* indice courant */
    int MIN;   /* position du minimum */
    int MAX;   /* position du maximum */
    /* Saisie des données */
    printf("Dimension du tableau (max.50) : ");
    scanf("%d", &N);
    for (I=0; I<N; I++)
    {
        printf("Elément %d : ", I);
        scanf("%d", &A[I]);
    }
    /* Affichage du tableau */
    printf("Tableau donné :\n");
    for (I=0; I<N; I++)
        printf("%d ", A[I]);
    printf("\n");
    /* Recherche du maximum et du minimum */
    MIN=0;
    MAX=0;
    for (I=0; I<N; I++)
    {
        if(A[I]>A[MAX]) MAX=I;
        if(A[I]<A[MIN]) MIN=I;
    }
    /* Edition du résultat */
    printf("Position du minimum : %d\n", MIN);
    printf("Position du maximum : %d\n", MAX);
    printf("Valeur du minimum : %d\n", A[MIN]);
    printf("Valeur du maximum : %d\n", A[MAX]);
    return 0;
}
```

Exercice 4

```
#include <stdio.h>
#include <math.h>
main()
{
    /* Prototypes des fonctions appelées */
    double F(int X);
    /* Variables locales */
    int I;
    /* Traitements */
    printf("\tX\tF(X)\n");
    for (I=1; I<=10; I++)
        printf("\t%d\t%f\n", I, F(I));
}
```

```

        return 0;
    }
    double F(int X)
    {
        return sin(X)+log(X)-sqrt(X);
    }

```

Exercice 5

```

#include <stdio.h>

main()
{
    /* Prototypes des fonctions appelées */
    double MIN(double X, double Y);
    double MAX(double X, double Y);
    /* Variables locales */
    double A,B,C,D;
    /* Traitements */
    printf("Introduire 4 réels : ");
    scanf("%lf %lf %lf %lf", &A, &B, &C, &D);
    printf("Le minimum des 4 réels est %f\n",
           MIN( MIN(A,B), MIN(C,D) ));
    printf("Le maximum des 4 réels est %f\n",
           MAX( MAX(A,B), MAX(C,D) ));
    return 0;
}

```

```

double MIN(double X, double Y)
{
    if(X<Y)
        return X;
    else
        return Y;
}

```

```

double MAX(double X, double Y)
{
    if(X>Y)
        return X;
    else
        return Y;
}

```

Exercice 6

Fonction minimum (entier tableau[], entier n)

```

    entier i;
    entier mini,maxi;
    mini ← tableau[0];
    maxi ← tableau[0];
    pour i allant de 1 à n
        si ( tableau[i]<mini)
            mini ← tableau[i];
        sinon si ( tableau[i]>maxi)
            maxi ← tableau[i];
    renvoyer mini et maxi

```

Exercice 7

```
double puiss (double x, int n) {  
    double resultat;  
    for (resultat = 1; n > 0; n--) resultat*=x;  
    return resultat;  
}  
  
/* Le programme de test */  
int main () {  
    double x; /* la valeur à fournir par l'utilisateur */  
    int n; /* la puissance à fournir par l'utilisateur */  
  
    /* Demande et lecture d'une valeur et d'une puissance */  
    printf("Introduire x : ");  
    scanf("%lf", &x);  
    printf("Introduire n : ");  
    scanf("%d", &n);  
  
    /* Affichage du résultat */  
    printf("%f^%d = %f\n", x, n, puiss(x, n));  
}
```

