

o

## Correction TD1 programmation 1

### Exercice 1 :

|                      |                        |
|----------------------|------------------------|
| i=i+f= .....         | f=i+f= .....           |
| i/j= .....           | i/f= .....             |
| f=i/j= .....         | j=i/f= .....           |
| f=(float) i/j= ..... | f=(float) (i/j)= ..... |

### Exercice 2 :

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int A,B,C,AIDE;
    printf("Introduisez trois nombres (A, B, C) : ");
    scanf("%d %d %d", &A, &B, &C);
    /* Affichage à l'aide de tabulations */
    printf("A = %d\tB = %d\tC = %d\n", A, B, C);
    AIDE=A;
    A=C;
    C=B;
    B=AIDE;
    printf("A = %d\tB = %d\tC = %d\n", A, B, C);
    return EXIT_SUCCESS;
}
```

### Exercice 3 :

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    /* Afficher le signe du produit de deux entiers sans
       faire la multiplication */
}
```

```

int A, B;
printf("Introduisez deux nombres entiers :");
scanf("%d %d", &A, &B);

if ((A>0 && B>0) || (A<0 && B<0))
    printf("Le signe du produit %d * %d est positif\n", A,B);

else if ((A<0 && B>0) || (A>0 && B<0))
    printf("Le signe du produit %d * %d est négatif\n", A,B);
else
    printf("Le produit %d * %dest zéro\n", A,B);
return EXIT_SUCCESS;
}

```

#### **Exercice 4 :**

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int N;    /* nombre de termes à calculer */
    int I;    /* compteur pour la boucle */
    float SOM; /* Type float à cause de la précision du résultat. */
    do
    {
        printf ("Nombre de termes: ");
        scanf ("%d", &N);
    }
    while (N<1);
    for (SOM=0.0, I=1;I<=N;I++)
        SOM += (float)1/I;
    printf("La somme des %d premiers termes est %f \n",N,SOM);
    return EXIT_SUCCESS;
}

```

#### **Exercice 5 :**

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int X,N; /* Les données */
    int I;   /* Le compteur */
    double RESU; /* Type double à cause de la */
               /* grandeur du résultat. */

    do
    {

```

```

    printf("Entrez l'entier naturel X : ");
    scanf("%d",&X);
}
while (X<0);
do
{
    printf("Entrez l'exposant      N : ");
    scanf("%d",&N);
}
while (N<0);

/* Pour N=0, le résultat sera automatiquement X^0=1 */
for (RESU=1.0,I=1;I<=N;I++)
    RESU*=X;

/* Attention: Pour X=0 et N=0 , 0^0 n'est pas défini */
if (N==0 &&X==0)
    printf("zéro exposant zéro n'est pas défini !\n");
else
    printf("Résultat : %d ^ %d = %.0f\n",X,N,RESU);
return EXIT_SUCCESS;
}

```

### **Exercice 6 :**

```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
int main()
{
    /* Calcul des solutions réelles d'une équation du second degré */
    int A, B, C;
    double D; /* Discriminant */
    printf("Calcul des solutions réelles d'une équation du second \n");
    printf("degré de la forme  ax^2 + bx + c = 0 \n\n");
    printf("Introduisez les valeurs pour a, b, et c : ");
    scanf("%d %d %d",&A,&B,&C);

    /* Calcul du discriminant b^2-4ac */
    D = pow(B,2) - 4.0*A*C;

    /* Distinction des différents cas */
    if (A==0 && B==0&&C==0) /* 0x = 0 */
        printf("Tout réel est une solution de cette équation.\n");
    else if (A==0 && B==0) /* Contradiction: c # 0 et c = 0 */
        printf("Cette équation ne possède pas de solutions.\n");
    else if (A==0) /* bx + c = 0 */
    {
        printf("La solution de cette équation du premier degré est :\n");
        printf(" x = %.4f\n", (double)C/B);
    }
}

```

```

    }
else if (D<0) /* b^2-4ac < 0 */
    printf("Cette équation n'a pas de solutions réelles.\n");
else if (D==0) /* b^2-4ac = 0 */
{
    printf("Cette équation a une seule solution réelle :\n");
    printf(" x = %.4f\n", (double)-B/(2*A));
}
else /* b^2-4ac > 0 */
{
    printf("Les solutions réelles de cette équation sont :\n");
    printf(" x1 = %.4f\n", (-B+sqrt(D))/(2*A));
    printf(" x2 = %.4f\n", (-B-sqrt(D))/(2*A));
}
return EXIT_SUCCESS;
}

```

### Exercice 7 :

a) avec la boucle while,

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    int N;    /* La donnée */
    int I;    /* Le compteur */
    double FACT; /* La factorielle N! - Type double à */
                /* cause de la grandeur du résultat. */

    do
    {
        printf("Entrez un entier naturel : ");
        scanf("%d", &N);
    }
    while (N<0);

    /* a */
    /* Pour N=0, le résultat sera automatiquement 0!=1 */
    I=1;
    FACT=1;
    while (I<=N)
    {
        FACT*=I;
        I++;
    }

    printf ("%d! = %.0f\n", N, FACT);
    return EXIT_SUCCESS;
}

```

```
}
```

### **b) avec la boucle for**

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int N;    /* La donnée */
    int I;    /* Le compteur */
    double FACT; /* La factorielle N! - Type double à */
                /* cause de la grandeur du résultat. */

    do
    {
        printf("Entrez un entier naturel : ");
        scanf("%d",&N);
    }
    while(N<0);

    for (FACT=1.0,I=1;I<=N;I++)
        FACT*=I;

    printf ("%d! = %.0f\n",N,FACT);
    return EXIT_SUCCESS;
}
```

### **Exercice 8 :**

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    /* Déclarations */
    int T[20]; /* tableau donné */
    int N;    /* dimension */
    int I;    /* indice courant */
    long SOM; /* somme des éléments - type long à cause */
                /* de la grandeur prévisible du résultat. */

    /* Saisie des données */
    printf("Dimension du tableau (max.20) : ");
    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");
/* Calcul de la somme */
for (SOM=0,I=0;I<N;I++)
    SOM += T[I];
/* Edition du résultat */
printf("Somme de éléments : %ld\n",SOM);
return EXIT_SUCCESS;
}

```

### **Exercice 9 :**

```

#include<stdio.h>
#include<stdlib.h>

int main()
{
    /* Déclarations */
    int U[20],V[20]; /* tableaux donnés */
    int N; /* dimension */
    int I; /* indice courant */
    long PS; /* produit scalaire */
    /* Saisie des données */
    printf("Dimension des tableaux (max.20) : ");
    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 : %ld\n", PS);
    return EXIT_SUCCESS;
}

```

### **Exercice 10 :**

```

#include <stdio.h>
#include <stdlib.h>

int 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 EXIT_SUCCESS;
}

```

**Exercice 11 :**

```

#include <stdio.h>
#include <stdlib.h>

int main()
{
    /* Déclarations */
    int T[20][20]; /* tableau donné */
    int L, C; /* dimensions */
    int I, J; /* indices courants */
    long SOM; /* somme des éléments - type long à cause */
                /* de la grandeur prévisible du résultat. */

    /* Saisie des données */
    printf("Nombre de lignes (max.20) : ");
    scanf("%d", &L );
    printf("Nombre de colonnes (max.20) : ");
    scanf("%d", &C );
    for (I=0; I<L; I++)
        for (J=0; J<C; J++)
        {
            printf("Elément[%d][%d] : ",I,J);
            scanf("%d", &T[I][J]);
        }
    /* Affichage du tableau */
    printf("Tableau donné :\n");
    for (I=0; I<L; I++)
    {
        for (J=0; J<C; J++)
            printf("%7d", T[I][J]);
        printf("\n");
    }
    /* Calcul de la somme */
    for (SOM=0, I=0; I<L; I++)
        for (J=0; J<C; J++)
            SOM += T[I][J];
    /* Edition du résultat */
    printf("Somme des éléments : %ld\n", SOM);
    return EXIT_SUCCESS;
}

```