

Voici le corrigé de l'exercice 1.

Pour la première question, il est question de l'étude sommaire de printf, scanf. La structure d'un programme est supposée connue. Référez-vous à la page suivante : [corrige_TP1](#)

La suite de l'exercice est :

◇ [Ecrivez un programme qui demande à l'utilisateur de saisir deux valeurs entières relatives puis affiche leur somme.](#)

Solution

```
#include<stdio.h>

int main( void )
{   int A,B,dif ;

    printf("Entrez A : ");
    scanf("%d", &A ) ; // adresse A = &A qui est transmis
au scanf

    printf("Entrez B : ");
    scanf("%d", &B ) ; // adresse B = &B qui est transmis
au scanf

    printf("La somme est : %d \n", A+B);

    return 0;
}
```

Pas grand chose à remarquer sur cet exercice, si ce n'est que l'on peut calculer une somme, de manière générale, on peut effectuer une opération dans un printf.

◇ [Ecrivez un programme qui demande à l'utilisateur de saisir deux valeurs entières relatives A et B puis affiche leur différence. Complétez ce programme pour qu'il affiche l'un des trois messages suivant le cas :](#)

"La différence A - B est plus grande que A"
"La différence A - B est plus petite que A"
"La différence A - B est nulle".

Solution :

```
#include<stdio.h>

int main( void )
{   int A,B,dif ;

    printf("Entrez A : ");
    scanf("%d", &A ) ; // adresse A = &A qui est transmis
au scanf

    printf("Entrez B : ");
    scanf("%d", &B ) ; // adresse B = &B qui est transmis
au scanf

    dif = A - B ; // = c'est une affectation !

    if ( dif > A )
        printf("La difference %d - %d est plus grande que
%d \n",A,B,dif);

    if ( dif < A )
        printf("La difference %d - %d est plus petite que
%d \n",A,B,dif);

    if ( dif == 0 ) // c'est == parce que c'est une
comparaison !
        printf("La difference %d - %d est nulle \n",A,
B);

    return 0;
}
```

Pas de difficultés particulières, juste ne pas oublier le double égal

pour la comparaison, voir la structure de l'instruction **if** (ici, pas de **else**, ni d'accolades vu que dans chaque cas il n'y a qu'une instruction à réaliser).

◇ Complétez le programme précédent pour qu'il saisisse des valeurs réelles. L'affichage des valeurs dans les trois messages se fera alors sur 8 caractères et 3 décimales.

Solution :

```
#include<stdio.h>

int main( void )
{   float A,B,dif ;

    printf("Entrez A : ");
    scanf("%f", &A ) ; // adresse A = &A qui est transmis
au scanf

    printf("Entrez B : ");
    scanf("%f", &B ) ; // adresse B = &B qui est transmis
au scanf

    dif = A - B ; // = c'est une affectation !

    if ( A > dif )
        printf("%La difference %8.3f - %8.3f est plus
grande que %8.3f\n",A,B,A);

    if ( A < dif )
        printf("%La difference %8.3f - %8.3f est plus
petite que %8.3f\n",A,B,A);

    if ( dif == 0 ) // c'est == parce que c'est une
comparaison !
        printf("La difference %8.3f - %8.3f est nulle
\n");
```

```
    return 0;
}
```

Peu de différences avec le programme précédent hormis le type des variables. On nous demande d'afficher sur au moins 8 caractères dont 3 réservées à la partie décimale d'où le %8.3f

Ecrivez un programme qui saisisse deux valeurs puis demande l'opération à effectuer : les réponses possibles seront +, -, * ou /. Vous utiliserez d'abord une série de test puis un switch pour choisir le type de calcul à effectuer.

Solution : (sans swicth)

```
#include<stdio.h>

int main( void )
{   float A,B,Res ;
    char choix;

    printf("Entrez A : ");
    scanf("%f", &A ) ; // adresse A = &A qui est transmis
au scanf

    printf("Entrez B : ");
    scanf("%f", &B ) ; // adresse B = &B qui est transmis
au scanf

    printf("Opération choisie : '+' , '-' , '*' , '/' :
");
    scanf(" %c", &choix ) ; // adresse choix = &choix qui
est transmis au scanf

    if ( choix == '+' )
        Res = A+B ;

    if ( choix == '-' )
        Res = A-B ;
```

```

    if ( choix == '*' )
        Res = A*B ;

    if ( choix == '/' )
        Res = A/B ;

    printf("Le résultat est : %8.3f\n",Res);

    return 0;
}

```

Pas de remarques très particulières, juste noter l'espace devant le %c dans le scanf(" %c", &choix). Cet espace sert à demander au système de supprimer les caractères présents dans le tampon d'entrée du clavier avant de procéder au scanf. Cela évite de récupérer les caractères correspondant à la frappe de la touche entrée ayant validé la saisie de B. Sans cet espace, le programme utilise cette valeur et passe à la suite sans attendre de saisie. Noter aussi que char étant un caractère, il faut comparer avec des caractères qui sont notés entre cotes simples.

Le gros défaut de ce programme est que si l'on saisit autre chose que '+', '-', '*' ou '/' le programme ne retourne pas de message d'erreur.

Noter l'utilisation du format %8.3f pour afficher sur 8 caractères minimum et 3 décimales.

Solution : (avec swicth)

```

#include<stdio.h>
#define TRUE 1
#define FALSE 0

int main( void )

```

```

{   float A,B,Res ;
    int ok;
    char choix;

    printf("Entrez A : ");
    scanf("%f", &A ) ; // adresse A = &A qui est transmis
au scanf

    printf("Entrez B : ");
    scanf("%f", &B ) ; // adresse B = &B qui est transmis
au scanf

    printf("Opération choisie : '+' , '-' , '*' , '/' :
");
    scanf(" %c", &choix ) ; // adresse choix = &choix qui
est transmis au scanf

    ok = TRUE ;
    switch ( choix )
    {   case '+' : Res = A+B ; break ;
        case '-' : Res = A-B ; break ;
        case '*' : Res = A*B ; break ;
        case '/' : Res = A/B ; break ;
        default :
            ok = FALSE ;
    }

    if ( ok )
        printf("Le résultat est : %8.3f\n",Res);
    else
        printf("Entree erronée \n") ;

    return 0;
}

```

Ce programme mérite quelques commentaires.

Tout d'abord sur le [switch](#) (allez voir la syntaxe ici). Dans le cas présent, il ne faut pas que l'on exécute les instructions suivantes du

bloc, il y a donc un `break` avant chaque case.

Notez la définition de constantes globales `TRUE` et `FALSE` et le test `if (ok)`. `ok` est un "drapeau" qui signale si la saisie est valide ou non. `ok` est positionnée à `TRUE`, et n'est modifiée que si la valeur de choix conduit le `switch` à exécuter la partie `default`.

La syntaxe du `if` prévoit une expression logique. En C toute valeur différente de 0 est équivalente à `TRUE`. Donc tester `ok`, c'est à dire tester sa valeur revient à tester si `ok != 0`. Par convention on prend `TRUE` égal à 1, mais n'importe quelle autre valeur aurait fait l'affaire.