

TD 1 – mise en œuvre des fonctions en langage C

1. Déclarez, codez et testez la fonction **afficherMenu** du diaporama dans un fichier main.c .
2. Déclarez, codez et testez la fonction **additionReel** du diaporama dans un fichier main.c .
3. Reprendre les questions 1 et 2 mais en créant des fichiers bibliothèque (voir la procédure sur le site, dans la rubrique codeBlocks).
4. Complétez vos fichiers bibliothèque en créant une fonction nommée **afficherCarre** qui affichera le carré d'une valeur entière passée en paramètre.
5. Complétez vos fichiers bibliothèque en créant une fonction nommée **calculerCarre** qui retournera le carré d'une valeur entière passée en paramètre.
6. Complétez vos fichiers bibliothèque en créant une fonction nommée **calculerDiscriminant** qui retournera la valeur du discriminant dont les éléments a,b et c sont passé en paramètre.

Rappel :

Pour un polynôme de type : $ax^2 + bx + c = 0$

Le discriminant vaut : $\Delta = b^2 - 4ac$

Si $\Delta > 0$ il existe 2 racines distinctes :

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a}$$

$$x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

7. Complétez vos fichiers bibliothèque en créant une fonction nommée **afficherRacines** qui affichera la valeur des racines correspondant au polynôme dont les éléments a,b et c sont passé en paramètre. (la fonction **sqrt** de la bibliothèque **math.h** permet d'avoir la racine carrée d'un valeur passée en paramètre).
8. Complétez vos fichiers bibliothèque en créant une fonction nommée **afficherUneLigne** prenant en paramètre un entier et affichant une ligne composée d'autant de '*' que le nombre passé en paramètre.

Exemple :

afficherUneLigne(5) donnera : *****

9. Complétez vos fichiers bibliothèque en créant une fonction nommée **afficherUnCarre** prenant en paramètre un entier et affichant un carré ayant pour hauteur le nombre passé en paramètre.

Exemple :

afficherUnCarre(3) donnera :

* * *

* * *

* * *

10. Complétez vos fichiers bibliothèque en créant une fonction **afficherRectangle**.
11. Complétez vos fichiers bibliothèque en créant une fonction nommée **compterSymbole** prenant en paramètre une chaîne de caractères, ainsi qu'un caractère et retourne le nombre d'occurrences de ce caractère dans la chaîne.

12. Ecrivez une fonction ayant le prototype suivant:

```
int verifTerminaison(char *chaine, char fin[2]);
```

La fonction devra retourner -1 si la chaine ne se termine pas par les 2 caractères compris dans le tableau fin et 1 dans si la chaine se termine par les deux caractères du tableau fin. Le programme de test est fourni:

```
#define NBCARFIN 2
#define NBMAXCAR 50
int main()
{
    char mot[NBMAXCAR];
    char terminaison[NBCARFIN]={'e','r'};
    int finEnEr;

    printf("donner un mot : ");
    scanf("%s",mot);
    finEnEr=verifTerminaison(mot,terminaison);
    if (finEnEr==1)
    {
        printf("%s se termine bien en ER\n",mot);
    }
    else
    {
        printf("%s ne se termine pas en ER\n",mot);
    }

    return 0;
}
```

13. Ecrivez la fonction ayant le prototype suivant:

```
int nbCroix(unsigned char grille[LARGEUR][HAUTEUR]);
```

Cette fonction retourne le nombre de 'X' présent dans le tableau à deux dimensions passé en paramètre.

Le programme de test est sur la page suivante.

```

#define LARGEUR 15
#define HAUTEUR 9
void initGrillle(unsigned char grille[LARGEUR][HAUTEUR])
{
    for(y=0; y<HAUTEUR; y++)
    {
        for(x=0; x<LARGEUR; x++)
        {
            grille[x][y]='.';
        }
    }
}

void afficheGrille(unsigned char grille[LARGEUR][HAUTEUR])
{
    int x,y;
    unsigned char lettre='A';
    printf(" ");
    for(x=0; x<LARGEUR; x++)
    {
        printf(" %c",lettre++);
    }
    printf("\n");
    for(y=0; y<HAUTEUR; y++)
    {
        printf("%d ",y+1);
        for(x=0; x<LARGEUR; x++)
        {
            printf(" %c",grille[x][y]);
        }
        printf("\n");
    }
}

int main()
{
    unsigned char tab[LARGEUR][HAUTEUR];
    int x,y;
    unsigned char lettre;
    int chiffre;
    int nombreDeCroix;

    initGrillle(tab);
    do
    {
        afficheGrille(tab);
        printf("ajouter une croix : lettreChiffre (ex: b4)\n");
        printf("fin avec a-1 : ");
        scanf(" %c%d",&lettre,&chiffre);
        if ((lettre>='a' && lettre<='a'+LARGEUR) && (chiffre>=1 && chiffre<=HAUTEUR))
        {
            tab[lettre-'a'][chiffre-1]='X';
        }
    }while (lettre!='a' || chiffre!=-1);
    nombreDeCroix=nbCroix(tab);
    printf("il y a %d croix \n",nombreDeCroix);
}

```