1 Traveaux Pratique 1

1.1 Programme affichant "Bonjour"

1.1.1 Code

```c
#include <stdlib.h>
#include <stdio.h>
void main()
{
    printf("Hello World");
}
```

Figure 1: Programme C pour le programme Hello World

Pour afficher "Hello World" nous devons simplement placer un *printf*.

1.1.2 Résultat

```
./GIMENEZ_bonjour
Hello World
```

Figure 2: Résultat du Programme *bonjour*

1.2 Calcul Aire d'un disque

1.2.1 Code

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

int main()
{
    int D;
    float pi = 3.14;

    printf ("Quel est le diamètre du disque ?\n");
    //D prend la valeur qu'entre l'utilisateur
    scanf("%d",&D);
    //On a : Aire = 2*pi*R². On effecture directement le calcul dans le printf pour faire l'économie d'une variable
    printf("La surface d'un cercle de rayon %d est : %.3f\n",D,(pi*D*D)/4);
    return 0;
}
```

Figure 3: Programme C pour le calcul de l'aire d'un Disque

Ici nous avons demandé le diamètre d'un disque pour pouvoir calculer son aire. Nous avons aussi affiché le résultat avec 3 chiffres après la virgule. Nous avons aussi fait le calcul directement dans le *printf*, pour faire l'économie d'une variable et rendre le programme plus lisible.

```
./GIMENEZ_disque
Quel est le diamètre du disque ?
2
La surface d'un cercle de rayon 2 est : 3.140
```

Figure 4: Résultat du Programme *disque*

1.2.2 Résultat

1.3 Signe du produit

1.3.1 Code

```c
#include <stdio.h>
int main()
{
    float a,b;
    //On entre les deux nombres dont on veut connaître le signe du produit
    printf ("Quel est le premier nombre ?\n");
    scanf("%f",&a);
    printf ("Quel est le deuxième nombre ?\n");
    scanf("%f",&b);

    //Traitement du cas particulier du produit nul
    if((a==0) || (b==0))
        printf("Produit nul\n");
    else
    {
        //Si A est positif le signe dépendra de b
        if (a>0)
            if(b>0)
                printf("Produit positif\n");
            else
                printf("Produit négatif\n");
        //Si A est négatif le signe dépend toujours de b mais les conditions sont inversée
        else
            if(b>0)
                printf("Produit négatif\n");
            else
                printf("Produit positif\n");
    }
    return 0;
}
```

Figure 5: Programme C pour la déterminaion du signe d'un produit

Nous cherchons à determiner si le signe résultant du produit de deux nombres sera positif ou négatif. Pour cela nous testons tout d'abord le cas particulier où l'un des deux nombre est nul. Par la suite en fonction du signe des nombres on pourra déterminer le signe du résultat.

1.3.2 Résultat

```
./GIMENEZ_produit
Quel est le premier nombre ?
1
Quel est le deuxième nombre ?
-1
Produit négatif
```

Figure 6: Résultat du Programme *produit*

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

int main()
{
    float x;
    //Saisi d'une valeur à donner à la variable pour déterminer f(x)
    printf ("Entrer un réel pour calculer la valeur de f(x) ?\n");
    scanf("%f",&x);
    //On calcul f(x) directement dans le printf, et on affiche une précision de 3 chiffres
    //après la virgule
    printf("f(%.3f) est : %.3f\n",x,3*pow(x,4)+2*pow(x,3)+pow(x,2)-x-2);


    return 0;
}
```

Figure 7: Programme C pour le calcul d'une fonction

1.4 Polynôme

1.4.1 Code

Nous souhaitons créer un programme permettant de calculer la valeur de la fonction f pour un x donné. Nous avons utilisé la fonction *power* de la librairie math.h, pour effectuer les calculs de puissance.

1.4.2 Résultat

```
./GIMENEZ_polynomes
Entrer un réel pour calculer la valeur de f(x) ?
2
f(2.000) est : 64.000
```

Figure 8: Résultat du Programme *polynome*

1.5 Conversion de secondes vers le format HH:MM:SS

1.5.1 Code

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

int main()
{
int sec_init;
int sec,min,hour;

//On entre le nombre de secondes que l'on souhaite convertir
printf("Entrer le nombre de secondes à convertir \n");
scanf("%d",&sec_init);

//On calcule combien d'heure cela représente
hour    = sec_init/3600;
//On soustrait le nombre d'heure pour trouver le nombre de minutes qu'il reste à convertir
min     = (sec_init-(3600*hour))/60;
//Enfin on soustrait le nombre de minutes et d'heure pour obtenir les secondes restantent
sec     = sec_init-((60*min)+(3600*hour));
//On affiche le résultat sous la format HH:MM:SS
printf("%d:%d:%d",hour,min,sec);

return 0;
}
```

Figure 9: Programme C pour la conversion d'un nombre de secondes en nombres d'heures, minutes et secondes

Nous avons ici créer un programme qui prend en entrées un nombre de secondes et qui l'a converti en nombres d'heures, de minutes, et de secondes. On notera que pour ce programme toute les divsions sont des divisions entière où on laisse le soin au compilateur d'arrondir les valeurs.

1.5.2 Résultat

```
./GIMENEZ_conversion
Entrer le nombre de secondes à convertir
13604
3:46:44
```

Figure 10: Résultat du Programme *conversion*

1.6 Calculatrice

1.6.1 Code

```c
int main()
{
    float A,B,R;
    int op, choix = 1;
    printf("Bienvenue sur le programme de calcul\n");
    //On boucle tant que l'utilisateur souhaite effectuer des calculs
    while (choix == 1)
    {
        //Entrée du premier nombre
        printf("Entrer un nombre\n");
        scanf("%f",&A);
        //Choix de l'opération
        printf("Quel opération souhaitez vous faire? (+-1 ou *-2)\n");
        scanf("%d",&op);
        //Si l'utilisateur à commis une erreur sur son choix on n'effectue pas le calcul
        if((op == 1) || (op == 2))
        {
            //Entrée du deuxième nombre
            printf("Entrer un deuxième nombre\n");
            scanf("%f",&B);
            //On calcul A * B ou A + B en fonction du choix de l'utilisateur et on affiche le résultat
            if (op == 1)  {R = A + B; printf ("%.3f + %.3f = %.3f\n",A,B,R); }
            else          {R = A * B; printf ("%.3f * %.3f = %.3f\n",A,B,R); }

        }
        //Si l'utilisateur s'est trompé on lui indique
        else printf("Erreur sur le choix..\n");
        //On demande si l'utilisateur souhaite continuer à utiliser le programme ou non
        printf("Souhaitez vous faire une autre opération? (1 = oui)\n");
        scanf("%d",&choix);
    }

    return 0;
```

Figure 11: Programme C pour la création d'une calculatrice

Ici nous avons créé un programme qui nous permet d'entrer deux nombres et d'effectuer une opération mathématique (+ ou *). Nous demandons alors à l'utilisateur d'entrer un premier nombre puis un opération et enfin un deuxième nombre. On vérifie que l'utilisateur à bien un opération valable.

```
./GIMENEZ_calculatrice
Bienvenue sur le programme de calcul
Entrer un nombre
5
Quel opération souhaitez vous faire? (+-1 ou *-2)
1
Entrer un deuxième nombre
12
5.000 + 12.000 = 17.000
Souhaitez vous faire une autre opération? (1 = oui)
1
Entrer un nombre

14
Quel opération souhaitez vous faire? (+-1 ou *-2)
5
Erreur sur le choix..
Souhaitez vous faire une autre opération? (1 = oui)
2
```

Figure 12: Résultat du Programme *calculatrice*

1.6.2 Résultat

2 Traveaux Pratique 2

2.1 Factorielle

2.1.1 Code

```c
int factorielle (int n)
{
    int i, R;
    //On initialise la variable résultat à 1 qui l'élément neutre du produit
    R = 1;
    for (i=2; i <= n; i++)
        R = R * i;
    return R;
}

int main()
{
    int i, max;
    max = 10;

    for (i = 0; i <= max; i++)
    {
        printf("Le resultat de %d! = %d\n",i,factorielle(i));
    }

    return 0;
}
```

Figure 13: Programme C du calcul de la facotrielle

Le but de ce code était de pouvoir calculer la factorielle d'un nombre et aussi d'afficher les résultats intermédiaires. On a ici créer une fonction pour faciliter la construction du texte. On a donc plus qu'a récupérer les valeurs et les afficher. La construction de la fonction fait que pour 0!, on obtient bien 1 comme résultat.

2.1.2 Résultat

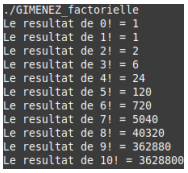

Figure 14: Résultat du Programme *factorielle*