

FICHER MAIN.C:

```
/**
 * @file main.c
 * @author Duradn Thomas
 * @brief fonction principale avec un menu pour choisir la fonction
 * appele
 * @version 0.1
 * @date 2019-10-14
 *
 * @copyright Copyright (c) 2019
 *
 */
#include <stdio.h>
#include <stdlib.h>

#include "plusGrand.h"
#include "partie2.h"

int main(){
    printf("bonjour que puis-je faire pour vous?\n");
    printf("1 pour table de multiplication\n");
    printf("2 pour le solveur d'equation\n");
    printf("3 pour la factoriel\n");
    printf("4 pour le sapin\n");
    printf("5 pour le rang de la factorielle\n");
    printf("6 pour l'écriture binaire\n");
    printf("7 pour le coef binomiale\n");
    printf("8 pour savoir si un nombre est Armstrong\n");

    int rep = saisiEnier();
    int res;

    switch (rep)
    {
    case 1:
        printf("rentrez un numero \n");
        tabMultiplication(saisiEnier());
        break;
    case 2:
        solEquation();
        break;
```

```

    case 3:
        printf("rentrez un numero \n");
        long long int fac = factoriel(saisiEnier());
        printf("le resultat est : %lld\n", fac);
        break;
    case 4:
        printf("quel longueur tu veux? \n");
        afficheSapin(saisiEnier());
        break;
    case 5:
        printf("quel rang tu veux? \n");
        res = maxFact(saisiEnier());
        printf("la reponse est: %d\n", res);
        break;
    case 6:
        printf("rentre un nombre\n");
        binaire(saisiEnier());
        break;
    case 7:
        printf("rentre n: ");
        int n = saisieEnier();
        printf("rentre p: ");
        int p = saisieEnier();
        res = coefBinomiale(n, p);
        printf("le resultat est : %d", res);
        break;
    case 8:
        printf("rentre un nombre ");
        estAmstrong(saisiEnier());
        break;

    default:
        printf("deole je n'ai pas ce que vous me demandez\n");
        break;
}

}

```

FICHIER partie2.c

```
/**
 * @file partie2.c
 * @author Durana Thomas
 * @brief ensemble des fonction de la 2eme partie du td
 * @version 0.1
 * @date 2019-10-14
 *
 * @copyright Copyright (c) 2019
 *
 */
#include <stdio.h>
#include <stdlib.h>

#include "plusGrand.h"

/**
 * @brief affiche un jolie sapin de n de longueur
 *
 * @param n
 */
void afficheSapin(int n){
    if (n ≤ 1){
        n = 2;
    }
    int debut = n - 1;
    int fin = n + 1;

    for (int i = 0; i < n; i++){
        for (int j = 0 ; j < debut; j++ ){
            printf(" ");
        }
        for (int j = debut ; j < fin-1; j++ ){
            printf("*");
        }
        for (int j = fin ; j < n; j++ ){
            printf(" ");
        }
        printf("\n");
        debut--;
        fin++;
    }
    debut = n - 2;
    fin = n + 2;
```

```

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0 ; j < debut; j++ ){
            printf(" ");
        }
        for (int j = debut ; j < fin-1; j++ ){
            printf("@");
        }
        for (int j = fin ; j < n; j++ ){
            printf(" ");
        }
        printf("\n");
    }
}

/**
 * @brief affiche le rang auquel la factoriel est superieur a k
 *
 * @param k
 * @return int
 */
int maxFact(int k){
    int facto = 0;
    int rang = 0;
    do{
        rang++;
        facto = factoriel(rang);
    } while (facto < k);
    return rang;
}

/**
 * @brief affiche l'écriture binaire d'un nombre
 *
 * @param n
 */
void binaire(int n){
    int k;
    for (int i = 31; i ≥ 0; i--)
    {
        k = n >> i;
        if (k & 1){
            printf("1");
        }
        else{

```

```

        printf("0");
    }
}
printf("\n");
}

```

```

/**
 * @brief calcule le coefficient binomiale avec n et p
 *
 * @param n
 * @param p
 * @return int
 */
int coefBinomiale(int n,int p){
    if (n == p || n < p){
        return 0;
    }else{
        int res;
        res = factoriel(n)/(factoriel(n-p)*factoriel(p));
        return res;
    }
}

```

```

/**
 * @brief verifie si un nombre est Amstrong ou pas
 * @param n
 */
void estAmstrong(int n){

    int test = n;
    int i = 1;
    int pos = 0;
    int aAjouter;
    int res=0;

    while (i < n){
        i = i * 10;
    }

    while (n > 0){
        aAjouter = n/i;
        n = n - (aAjouter*i);
        res += aAjouter * aAjouter * aAjouter ;
        pos++;
    }
}

```

```
    i = i / 10;
}
if (res == test){
    printf("oui il l'est \n");
}else{
    printf("non il ne l'est pas \n");
}
}
```

## FICHIER PLUSGRAND.C

```
/**
 * @file plusGrand.c
 * @author Durana Thomas
 * @brief ensemble des fonction demande dans la partie 1 du Tp3
 * @version 0.1
 * @date 2019-10-14
 *
 * @copyright Copyright (c) 2019
 *
 */
#include <stdio.h>
#include <stdlib.h>

/**
 * @brief demande un int a l'utilisateur
 *
 * @return int
 */
int saisiEnier(){
    int n,test;
    test = scanf("%d",&n);
    if (test == 1){
        return n;
    }else{
        printf("rentre un vrai numero stp chef \n");
        exit(EXIT_SUCCESS);
    }
}

/**
 * @brief affiche la table de multiplication d'un entier demande
 *
 * @param n
 */
void tabMultiplication(int n){
    for (int i = 1; i ≤ 10; i++)
    {
        printf("%d * %d = %d\n",i,n,i*n);
    }
}

/**
```

```

* @brief solveur d'equation par dicotomie
*
*/
void solEquation(){
    float calcule,x,a,b;
    a = -15;
    b = -10;

    do{
        x = (a + b)/2;
        calcule = x*x*x + 12*x*x + 1;
        if (calcule ≤ 0){
            a = (a+b)/2;
        }else{
            b = (a+b)/2;
        }
    } while (calcule > 0.00001 || calcule < -0.00001);
    printf("[%f,%f]\n",a,b);
}

/**
* @brief fonction factoriel recursive
*
* @param n
* @return int
*/
long long int factoriel(long long int n){
    if (n == 0){
        return 1;
    }else{
        return factoriel(n-1) * n;
    }
}

```