

Corrigé du TP6

Langage C

Y. ALJ

1 avril 2020

1 Corrigé des exercices

1.1 Exercice 1 :

Questions 1-2

```
// Questions 1 et 2
#include <stdio.h>
//declaration de la fonction puissance
int puissance(int valeur, int exposant);

//definition de la fonction main
void main(){
    // declaration des variables utilisees
    int val, exp, resultat;
    // on demande a l'utilisateur de saisir une valeur
    printf("val?\n");
    // on lit la valeur saisie
    scanf("%d", &val);
    // on demande a l'utilisateur de saisir un exposant
    printf("exposant?\n");
    // on lit l'exposant saisi
    scanf("%d", &exp);
    // on appelle la fonction puissance definie ci-dessous
    // le resultat du calcul de val a la puissance exp
    // sera stocke dans resultat
    resultat = puissance(val, exp);
    // on affiche le resultat trouver
    printf("resultat=%d\n", resultat);
}

//definition de la fonction puissance
int puissance(int valeur, int exposant){
    // Tres important!!! on initialise le resultat a 1
    // pour pouvoir faire la multiplication apres
    int resultat = 1;
    // initialisation de la variable de la boucle
    int i;
    // on fait une boucle car :
    // on sait que :  $a^n = a * a * \dots * a$  n-fois
    for (i = 0; i < exposant; i++){
        // equivalent a : resultat = resultat * valeur;
        resultat *= valeur;
    }
    // on renvoie le resultat pour pouvoir le recuperer dans l'appel de la fonction
    return resultat;
}
```

Question 3

```
// Question 3
#include <stdio.h>
//declaration de la fonction puissance
int puissance(int valeur, int exposant);
//definition de la fonction main
void main(){
    // declaration des variables utilisees
    int val, exp, resultat, i;
    for (i=0; i<10; i++){
        // on demande a l utilisateur de saisir une valeur
        printf("val?\n");
        // on lit la valeur saisie
        scanf("%d", &val);
        // on demande a l'utilisateur de saisir un exposant
        printf("exposant?\n");
        // on lit l exposant saisi
        scanf("%d", &exp);
        // on appelle la fonction puissance definie ci-dessous
        // le resultat du calcul de val a la puissance exp
        // sera stocke dans resultat
        resultat = puissance(val, exp);
        // on affiche le resultat trouver
        printf("resultat=%d\n", resultat);
    }
}

//definition de la fonction puissance
int puissance(int valeur, int exposant){
    // Tres important!!! on initialise le resultat a 1
    // pour pouvoir faire la multiplication apres
    int resultat = 1;
    // initialisation de la variable de la boucle
    int i;
    // on fait une boucle car :
    // on sait que :  $a^n = a * a * \dots * a$  n-fois
    for (i = 0; i < exposant; i++){
        // equivalent a : resultat = resultat * valeur;
        resultat *= valeur;
    }
    // on renvoie le resultat pour pouvoir le recuperer dans l appel de la fonction
    return resultat;
}
```

1.2 Exercice 2 :

```
#include<stdio.h>
// pour sqrt et pour pow
#include<math.h>

// declaration de distance
double distance(double xa, double ya, double xb, double yb);

// definition de la fonction main
void main(){
    double xa, ya, xb, yb, d;
    // on demande a l'utilisateur de saisir des valeurs
    // des coordonnees A et B
    printf("saisissez les coordonnees x et y de A?\n");
    scanf("%lf %lf", &xa, &ya);
    printf("saisissez les coordonnees x et y de B?\n");
    scanf("%lf %lf", &xb, &yb);
    // appel de la fonction distance
    d = distance(xa, ya, xb, yb);
    printf("la distance AB est %.2lf", d);
}

// definition de la fonction distance
double distance(double xa, double ya, double xb, double yb){
    return sqrt(pow((xb-xa),2) + pow((yb-ya),2));
}
```

1.3 Exercice 3 :

Questions 1-2

```
// exo3 question 1 et 2
#include <stdio.h>
// declaration des fonctions fact, affiche_fact
int fact(int n);
void affiche_fact(int n);

// definition de la fonction main
void main(){
    int n;
    printf("saisissez un entier positif ou nul\n");
    scanf("%d", &n);
    //appel de la fonction affiche_fact
    affiche_fact(n);
}

// definition de la fonction fact
int fact(int n){
    int res = 1;
    int i;
    for (i=1; i <= n; i++){
        res *= i;
    }
    return res;
}

// definition de la fonction affiche_fact
void affiche_fact(int ent){
    int factorielle;
    // appel de la fonction factorielle
    factorielle = fact(ent);
    printf("la factorielle de %d est %d\n", ent, factorielle);
}
```

Question 3

```
//exo3 question 3
#include <stdio.h>
// declaration des fonctions fact , affiche_fact , comb et affiche_comb
int fact(int n);
void affiche_fact(int n);
int comb(int n, int p);
void affiche_comb(int n, int p);

// definition de la fonction main
void main(){
    int n, p;
    printf("saisissez n : un entier positif ou nul\n");
    scanf("%d", &n);
    printf("saisissez p : un entier plus petit ou egal a n\n");
    scanf("%d", &p);
    affiche_comb(n,p);
}

// definition de la fonction fact
int fact(int n){
    int res = 1;
    int i;
    for (i=1; i <= n; i++){
        res *= i;
    }
    return res;
}

// definition de la fonction comb
int comb(int n, int p){
    int numerateur;
    int denominateur;
    if ( p > n){
        printf("erreur : p doit etre superieur ou egal a n");
        return -1;
    }
    else{
        // on utilise la formule  $C(n,p) = n! / p! * (n-p)!$ 
        numerateur = fact(n);
        denominateur = fact(p)*fact(n-p);
        return numerateur/denominateur;
    }
}

// definition de la fonction affiche_fact
void affiche_fact(int ent){
    int factorielle;
    // appel de la fonction factorielle
    factorielle = fact(ent);
    printf("la factorielle de %d est %d\n", ent, factorielle);
}

// definition de la fonction affiche_comb
void affiche_comb(int ent1, int ent2){
    int combinaison;
    // appel de la fonction factorielle
    combinaison = comb(ent1, ent2);
    printf("le nombre de combinaisons de %d elements parmi %d est %d\n", ent2, ent1,
        combinaison);
}
```