

DM1 - Corrigé

Hicham Janati

1 Exercices indépendants

Les exercices sont indépendants, vous pouvez tout de même les utiliser si vous jugez cela utile.

1. Écrire un programme *main* qui affiche les *n* premiers carrés parfaits où *n* est saisi par l'utilisateur

```
#include<stdio.h>
int main () {
    int n,i;
    printf("Saisir un entier n :"); scanf("%d", &n);
    for(i=1; i<=n; i++) {
        printf("Le carré de %i est %d\n", i, i*i);
    }
    return 0;
}
```

2. Écrire un programme *main* qui affiche le décor suivant en un nombre de lignes *n* saisi par l'utilisateur.

Exemple pour *n* = 4 :

```
*
**
***
****
```

```
#include<stdio.h>
int main () {
    int n,i,j;
    printf("Saisir le nombre de lignes n:\n");
    scanf("%d", &n);
    for(i=1; i<=n;i++) {
        for(j=1; j<=i; j++)    printf("*");
        printf("\n");
    }
    return 0;
}
```

3. Même question pour le décor suivant :

```
1
2 3
4 5 6
7 8 9 10
```

```

#include<stdio.h>
int main () {
    int n,i,j, c=0;
    printf("Saisir le nombre de lignes n :\n");
    scanf("%d", &n);
    for(i=1; i<=n;i++) {
        }
    for(j=0; j<i; j++) { c+=1; printf("%d ", c);
    } printf("\n");
    return 0;
}

```

4. **Erreur d'énoncé : oubli du mot angle.** Écrire une fonction qui prend en paramètres trois entiers et renvoie 1 s'ils peuvent être des mesures d'angles d'un triangle non plat, -1 si le triangle est plat, et 0 si elle ne peuvent pas former un triangle.

```

int Triangle(int A, int B, int C){
    int somme = A+B+C;
    // Si la somme des angles n'est pas 180, on ne peut pas construire un tel triangle
    if somme != 180:
        return 0;
    // Si l'un des angles est nul, le triangle est plat
    if A*B*C == 0:
        return -1;
    return 1;
}

```

5. Un nombre premier est divisible par 1 et lui-même uniquement. Écrire une fonction qui vérifie si un nombre est premier ou pas. (Elle renvoie 1 le cas échéant).

```

int Premier (int n) {
    int i;
    for(i=2;i<n;i++){
        if (n%i==0)
            return 0;
    }
    return 1;
}

```

6. Écrire une fonction qui compte le nombre de diviseurs d'un entier naturel.

```

int Diviseurs(int n) {
    int i, ndiv=0;
    for(i=1;i<=n; i++){
        if (n%i==0) ndiv+=1;
    }
    return ndiv;
}

```

7. Écrire une fonction qui compte le nombre de diviseurs premiers d'un entier naturel.

```

int DiviseursPremiers (int n){
    int i, ndivp=0;
    for(i=1;i<=n;i++){
        if(Premier(i) && n%i==0) ndivp++;
    }
}

```

```
return ndivp:
}
```

2 Problèmes

2.1 Un peu d'aléatoire

1. Écrire une fonction Colle qui prend 2 entiers en paramètres (a,b) et renvoie le nombre entier construit en "collant" les arguments "ab". Exemple : Colle(4,3) vaut 43. Colle(0,4) vaut 4. Colle(3,0) vaut 30.

```
int Colle(int a, int b){
    return 10*a + b;
}
```

2. En utilisant la fonction rand() qui renvoie un entier aléatoire, écrire une fonction qui renvoie un entier aléatoire entre 0 et 9.

```
int random(){
    return rand()%10;
}
```

3. Écrire une fonction qui fait deux tirages entre 0 et 9 (imaginez deux dés à 10 faces chacun) et les rassemble en un nombre et le renvoie. Si les trois tirages sont 3 et 1, la fonction renvoie 31.

```
int DeuxTirages(){
    int a = random(), b = random();
    return Colle(a,b);
}
```

4. Écrire une fonction qui fait autant de tirages de dés doubles à 10 faces qu'il n'en faut jusqu'à faire le tirage 2 et 8.

```
int TirageBizarre(){
    int d;
    do{
        d = DeuxTirages();
    }while(d != 28 && d != 82);
    return d;
}
```

2.2 Et ce n'est pas fini ...

On va jouer au loto! Un joueur dispose d'un montant d'argent initial M (qu'il saisit au début). Tant que son capital lui permet d'acheter un ticket à 5 euros, il peut jouer, sinon gameover! Il peut aussi quitter à tout moment et partir avec la somme dont il dispose.

Après l'achat du ticket, l'utilisateur doit saisir 10 entiers entre 1 et 50. Ces nombres sont stockés dans un tableau de taille 10.

Ensuite, on procède au tirage : 5 nombres sont tirés aléatoirement entre 1 et 50. Les récompenses sont comme suit :

- 0 nombres trouvés : 0 euros
- 1 nombre trouvé : +5 euros (il rentabilise le ticket)
- 2 nombres trouvés : +20 euros
- 3 nombres trouvés : +100 euros
- 4 nombres trouvés : +200 euros
- 5 nombres trouvés : +1 000 000 euros Jackpot! On arrête le jeu!

1. Écrire une fonction Loto qui génère un entier aléatoire entre 1 et 50.

2. Écrire une fonction Mise qui prend un tableau en paramètre de taille 10 et le remplit avec des nombres saisis par le joueur.
3. Écrire une fonction Tirage qui prend un tableau en paramètre de taille 5, remplit à l'aide de la fonction Loto et l'affiche en même temps.
4. Écrire une fonction Comptage qui prend en paramètre un tableau A et taille 5 (loto) et un tableau B de taille 10 (mise) et renvoie le nombre de valeurs communes entre les tableaux.
5. Écrire la fonction main qui fait le jeu : n'oubliez pas que le joueur joue tant qu'il a assez d'argent et qu'il VEUT jouer ! Voici une sortie d'une partie jouée, après 3 tours le joueur décide de s'arrêter :

Fichier fonctions.h

```
void Mise(int T[]);
void Tirage(int T[]);
int Comptage(int A[], int B[]);
```

Fichier fonctions.c

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

void Mise(int T[]){
    int i;
    for(i=0;i<10;i++){
        printf("Mise [%d]=",i);
        scanf("%d",&T[i]);
    }
}

void Tirage(int T[]){
    int i;

    for(i=0;i<5;i++){
        T[i] = rand()%50+1;
        printf(" %d \n",T[i]);
    }
}

int Comptage(int A[], int B[]){
    int i,j, compte=0;

    for(i=0;i<5;i++){
        for(j=0;j<10;j++){
            if (B[j] == A[i]) compte++;
        }
    }
    return compte;
}
```

Fichier main.c

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include"fonctions.h"
```

```

int main(){
    int envie=1;
    int M;
    int TabMise[10];
    int TabLoto[5];
    int resultat;

    printf("=====\n===== Bienvenue !=====\n");

    //Le joueur saisit le montant d'argent qu'il a
    printf("Veuillez saisir votre montant initial:");
    scanf("%d",&M);

    // On garde une copie du montant initial:
    int MontantInitial = M;

    // Tant qu'il peut acheter un ticket (M>5) et qu'il veut jouer (envie=1), on joue !
    while(envie && M>5){
        //Il paie son ticket à 5 euros, on retranche donc 5 à M:
        M -= 5;

        //Le joueur doit maintenant remplir le tableau:
        printf("\n===== Préparez-vous, un nouveau tour commence ! ===== \n");
        printf("\nVeuillez saisir les nombres de votre mise:\n");
        Mise(TabMise);

        printf("\n les nombres gagnants du tirage sont:\n");
        //On fait le tirage de 5 valeurs et on remplit le tableau TabLoto:
        Tirage(TabLoto);

        //On compte le nombre de bonnes prédictions:
        resultat = Comptage(TabLoto, TabMise);

        //On attribue les gains:
        if (resultat == 1) M+= 5;
        else if (resultat == 2) M += 20;
        else if (resultat == 3) M += 100;
        else if (resultat == 4) M += 200;
        else if (resultat == 5)
        {
            M += 1000000;
            printf("\n ===== JACKPOT =====\n !!! vous êtes millionnaire !!!\n");
            break; //on arrête la boucle
        }

        printf("Votre budget actuel est désormais :%d\n",M);

        //On demande si le joueur veut continuer, s'il saisit 0, envie deviendra 0 et la
        boucle s'arrête
        printf("Voulez-vous continuer à jouer (1/0) ?\n");
        scanf("%d",&envie);

        if (M<5 && envie) printf("Désolé ! vous ne pouvez plus acheter de ticket\n");
        if (envie !=1) printf("à la prochaine !\n");

    }
}

```

```
return 0;  
}
```