Timéo
Fayat

TP Langage C : Projet
Sujet A

Sommaire :

Introduction :

Pour ce projet de langage C, j'ai décidé de faire le sujet A par rapport à mon niveau dans cette matière. Dans ce sujet, l'objectif est de créer des exercices de mathématiques et de le faire sous une forme de petit jeu. Pour ce faire, nous devrons faire différents programmes pour différents calculs mathématiques, et afin de le rendre ludique, nous utiliserons un système de points pour récompenser l'utilisateur pour ses bonnes réponses.
Afin de réaliser ce projet, j'ai utilisé le logiciel CodeBlocks.

I. Début dirigé

1.

Pour cette première étape, il suffit d'écrire chaque printf pour chaque ligne, sans oublier les \n afin de passer une ligne à chaque fois :

```
int main() {
    int i;
    printf("+------------------------------+\n");
    printf("|1 : Addition                  |\n");
    printf("|2 : Soustraction              |\n");
    printf("|3 : Multiplication            |\n");
    printf("|4 : Tables des multiplications  |\n");
    printf("|5 : Divisions                 |\n");
    printf("|0 : Sortir du jeu             |\n");
    printf("+------------------------------+\n");
    printf("Quel est votre choix ?\n");
    return 0;
}
```

2.

Ici, on ajoute donc d'abord un scanf afin de demander ce que l'utilisateur souhaite exécuter parmis ces choix, sans oublier l'&.

```
scanf("%d", &i);
```

Ensuite, j'ai choisi d'utiliser un switch pour chaque valeur de i entre 0 et 5, avec une erreur si i n'est pas entre 0 et 5 :

```
switch (i) {
        case 1 : printf("Addition\n"); break;
        case 2 : printf("Soustraction\n"); break;
```

```
    case 3 : printf("Multiplication\n"); break;
    case 4 : printf("Table des multiplications\n"); break;
    case 5 : printf("Divisions\n"); break;
    case 0 : printf("Sortie\n"); break;
    default : printf("Erreur : vous devez choisir un chiffre entre 0 et 5");
  }
```

3.

Ici, on importe la fonction srand afin d'affecter par la suite un entier au hasard entre 0 et 100 à a et b. On demande le résultat de la somme à l'utilisateur puis le programme répond si la réponse est correct ou non :

```
case 1 : printf("Addition\n");
    srand (time(NULL));
    a=rand() % 101;
    b=rand() % 101;
    printf("Combien fait %d + %d \n", a, b);
    scanf("%d", &c);
    if (c==a+b){
       printf("Bravo");}
       else {
          printf("Raté");}
    break;
```

4.

On ajoute en début de programme une boucle while afin de répéter le, processus jusqu'à ce que l'utilisateur choisisse 0 :

```
 int i, a, b, c;
   i=1;
   while (i!=0){
```

5.

Déjà fait.

II.Des fonctions

1.

Afin de coder la fonction addition, on utilise d'abord void car la fonction ne contiendra pas de return. Pour le reste, il suffit de faire comme déjà fait pour la question précédente.

```
void addition() {
    srand(time(NULL));
    int a, b, c
    a = rand() % 101;
    b = rand() % 101;
    printf("Quel est la somme de %d et %d ?\n", a, b);
    scanf("%d", &c);
    if (c == a + b) {
        printf("Bravo !\n");
    } else {
        printf("Dommage ! La bonne réponse était %d.\n", a + b);
    }
}
```

2.

Pour la fonction multiplication, il suffit de remplacer le + par * dans la fonction addition.

```
void multiplication() {
    srand(time(NULL));
    int a, b, c;
    a = rand() % 10 + 1;
    b = rand() % 10 + 1;
    printf("Quel est le produit de %d et %d ?\n", a, b);
    scanf("%d", &c);
    if (c == a * b) {
        printf("Bravo !\n");
    } else {
        printf("Dommage ! La bonne réponse était %d.\n", a * b);
    }
}
```

3.

Ici la soustraction s'effectue de la même manière, sauf que l'on ajoute une inversion entre a et b si b est supérieur à a, pour cela nous utilisons une troisième variable que l'on utilisera pour garder a après l'avoir remplacé par b.

```
void soustraction(){
    int s, a, b, c;
    srand(time(NULL));
    a = rand() %101;
    b = rand() %101;
    if (a<b) {
        c=a;
        a=b;
        b=c;}
    printf("Quelle est la différence de %d et %d ?\n", a,b);
    scanf("%d", &s);
    if (s == a-b) {
        printf("Bravo !\n"); }
    else {
        printf("Dommage, la bonne réponse était %d \n", a-b);
    }
}
```

III. En autonomie

1.

Pour la gestion de points, on transforme les fonctions de telle sorte : “int addition(int pts)” pour chaque fonction, on retourne pts+1 si le résultat est bon et pts si non.
Ensuite on finit le programme en affichant le nombre de point
printf ("Vous avez %d points\n", pts);

2.

Ici la fonction ne retourne rien donc on utilise void, puis on demande à l’utilisateur quelle table veut-il, et enfin une boucle for pour afficher toute la table de l’entier choisi.

```
void table(){
    int a;
    printf("Quelle table voulez-vous entre 1 et 10 ?");
    scanf("%d", &a);
    for (int i=1; i<=10; i++){
        printf("%d\n", a*i);
        }
}
```

3.

Ici on modifie la fonction afin de calculer les points pour chaque bonne réponse. De plus, on ajoute une deuxième boucle for afin de demander le produit de l’entier choisi avec chaque chiffre entre 1 et 10. Ensuite un if pour vérifier que la réponse est correct afin de distribuer les points.

```
int table(int pts){
    int a, b;
    printf("Quelle table voulez-vous entre 1 et 10 ?");
    scanf("%d", &a);
    for (int i=1; i<=10; i++){
        printf("%d\n", a*i);
        }
    for (int i=1; i<=10; i++){
        printf("%d * %d = ?\n", a, i);
        scanf("%d", &b);
        if(b==a*i){
            printf("Gagné");
            pts=pts+1;
```

```
    }
        else{
            printf("Perdu");
        }
    }
return (pts);
}
```

IV. Gestion des points

1.

Pour sauvegarder les points :
On commence par créer une fonction qui nous permettra de récupérer la date et l'heure de l'ordinateur afin de simplifier le programme. Pour ce faire on déclare une variable de type time_t pour représenter le temps, ensuite on déclare un pointeur vers une structure qui nous servira à convertir l'heure en une structure plus détaillée. Ensuite, la fonction time(&t) récupère l'heure actuelle sur l'ordinateur. La fonction localtime(&t) prend en paramètre le time_t et retourne un pointeur vers la structure qui contient la date et l'heure, la valeur est ensuite stockée dans tm_info. La fonction strftime modifie la structure pour que la date soit sous la forme AAAA-MM-JJ. La même chose pour l'heure sous la forme HH: MM: SS.

```
#include <time.h>


void recupererDateHeure(char* date, char* heure) {
    time_t t;
    struct tm* tm_info;

    // Récupérer l'heure système actuelle
    time(&t);

    // Convertir l'heure en structure tm
    tm_info = localtime(&t);

    // Formater la date sous la forme "AAAA-MM-JJ"
    strftime(date, 20, "%Y-%m-%d", tm_info);

    // Formater l'heure sous la forme "HH:MM:SS"
    strftime(heure, 10, "%H:%M:%S", tm_info);}
```

Ensuite pour sauvegarder dans un fichier, on ouvre un fichier nommé FICHIERPTS en mode "a" pour écrire dans le fichier sans supprimer ce qui a été écrit précédemment. On écrit ensuite dans le fichier le nom, le nombre de point, et la date et l'heure récupéré grâce à la fonction recupererDateHeure.

```
FILE* fpts;
    fpts = fopen("FICHIERPTS", "a");
    char date[20];
    char heure[20];
    recupererDateHeure(date, heure);
    printf("Entrez votre nom"); scanf("%s", nom);
    fprintf(fpts,"%s  %d  %s  %s\n", nom, pts, date, heure);
    fclose(fpts);
```

Pour récupérer son ancien nombre de points grâce à notre nom, on crée une fonction pour le faire. La fonction prend en paramètre le nom de l'utilisateur puis récupère le fichier en mode lecture soit "r". Le while permet à ce que la chaîne de caractère donnees récupère chaque ligne, et pour chaque ligne, on vérifie si le nom écrit correspond au nom entré par l'utilisateur. Lorsque c'est le cas, on retourne le nombre de points, lorsque le nom n'est pas trouvé, la fonction s'arrête et ne retourne rien. Dans le programme, on écrira que si la fonction ne retourne rien, alors on envoie à l'utilisateur qu'aucun score n'a été sauvegardé pour ce nom.

```
int recupererScore(char* nom) {
    FILE* fpts = fopen("FICHIERPTS", "r");

    char donnees[100];
    char nomfichier[50];
    int pts;
    char date[20], heure[10];


    while (fgets(donnees, sizeof(donnees), fpts)) {
        sscanf(donnees, "%s %d %s %s", nomfichier, &pts, date, heure);
        if (strcmp(nomfichier, nom) == 0) {
            fclose(fpts);
            return pts;
        }
    }

    fclose(fpts);
    return -1;
}

printf("Entrez votre nom pour récupérer votre score: ");
    scanf("%s", nom);
    pts = recupererScore(nom);

    if (pts != -1) {
        printf("Votre score est : %d\n", pts);
    } else {
        printf("Aucun score trouvé pour ce nom.\n");
    }
```

2.

On reprend la fonction addition, si la première réponse est bonne on retourne pts+10, si elle est fausse, la fonction lance une boucle while qui ajoute 1 à chaque tentative et donne une nouvelle tentative à l'utilisateur. Enfin, un if afin de retourner pts+5 si la bonne réponse a été donnée à la deuxième tentative, si c'est plus, alors l'utilisateur n'a qu'1 point supplémentaire. Par la suite, nous appliquons ce processus à chaque fonction.

```
int addition(int pts) {
    srand(time(NULL));
    int a, b, c, e;
    a = rand() % 101;
    b = rand() % 101;
    printf("Quel est la somme de %d et %d ?\n", a, b);
    scanf("%d", &c);
    e=1;
    if (c == a + b) {
        printf("Bravo !\n");
        return (pts + 10);
    } else {
        while (c!=a+b) {
            printf("Dommage ! Essayez encore\n");
            e=e+1;
            scanf("%d", &c);
        }
        printf("Bravo !\n");
        if (e==2){
            return (pts+5);
        } else {
            return(pts+1);
    }
}
}
```

Difficultés rencontrées

Pour réaliser ce projet, j'ai rencontré beaucoup de difficultés lors des sauvegardes des points dans un fichier. En effet, j'ai d'abord peiné à réussir à aborder les fonctions liées aux fichiers, mais le plus compliqué a été l'étape de la récupération de points. Pour réussir cette étape, j'ai dû chercher sur différents sites internet d'explications du langage C. Je l'ai également fait pour aborder la fonction permettant de récupérer la date et l'heure. Après tout cela, je pense avoir plutôt bien réussi à combler mes lacunes liées aux sauvegardes.

Lien Github

https://github.com/TimeoFyt/Tim-o-Fayat.git