

# **RAPPORT DEVOIR1-AI5**

**Montée en compétences C**

**MBOUOPDA JEFFERSON**

**Mougoué Ruben**

**Novembre 2023**

TABLE DES MATIÈRES

EXERCICE 1..... 3

EXERCICE 2..... 6

EXERCICE 3..... 7

## EXERCICE 1

### Programme 1

```
#include <stdio.h>
int main () {
    int A=20,    B=5;
    int C=!--A/++! B;
    printf ("A=%d B=%d c=%d \n", A, B, C);
}
```

Ce programme ne peut être exécuté, car il y a une erreur à la ligne 4 du programme :

“ `int C=!-- A/++! B;` “. L’association des opérateurs de cette manière-là ne permet pas de retourner un résultat.

### Programme 2

```
#include <stdio.h>
int main () {
    int A=20,    B=5,    C=-10,    D=2;
    printf ("%d \n", A&&B || !0&&C++&&!D++);
    printf ("c=%d d=%d \n", C, D);
}
```

`A && B => 0` (Vraie en booléen)

`!0 && (C++) && !(D++) => 1` ( `C++=-10` différent de 0 et de 2)

Avec le OU de ces deux relations le premier printf renvoie 1.

Le deuxième printf renvoie les valeurs de C et D pas encore incrémentées d’où les valeurs -10 et 2 sont affichées.

### Programme 3

```
#include <stdio.h>
int main () {
    int p[4]={1, -2, 3, 4};
    int *q=p;
    printf ("c=%d\n",  *++q**q++);
    printf ("c=%d \n", *q);
}
```

Dans le premier printf, on a la valeur de `*++q**q++` qui s'affiche qui est ici 4, car `*++q = -2` multiplié par `*q++ = -2` donne 4.

Le deuxième printf affiche la valeur de `*q` qui est 3 parce que `*(q++)` est une opération post-incrémentation.

### Programme 4

```
#include <stdio.h>
int main () {
    int p[4]={1, -2, 3, 4};
    int *q=p;
    int d=*q&*q++|*q++;
    printf ("d=%d\n", d);
    printf ("q=%d \n", *q);
}
```

`*(q) & *(q++) => 1`

`*(q++) => -2`

Or `1 OU -2 => 1 + (-2) => d = -1` (premier printf)

Le deuxième printf affiche le contenu du pointeur q qui est 3, car d implémente deux fois la position pointée par q.

### Programme 5

```
#include <stdio.h>
int main () {
    int a=-8, b=3;
    int c= ++a && --b ? b-- : a++;

    printf ("a=%d b=%d c=%d\n", a, b, c);
}
```

On regarde si  $(++a) \&\&(--b)$  est vérifié ou non. Ici  $(++a) = -7$  ET  $(--b) = 2$  ce qui est vrai. Ainsi la condition “b - -” est réalisée. Donc  $c=2$ , car b - - est une opération post-incrémentation et  $b = 1$ .

## Programme 6

---

```
#include <stdio.h>
int main () {
    int a=-8, b=3;
    a >= 2^b;
    printf ("a=%d\n", a);
}
```

Résultat obtenu : a = -4

## EXERCICE 2

1. Ce programme permet de lire les notes de N étudiants de l'UTC dans un devoir de l'UV SR01 et les mémorise dans un tableau POINTS de dimension N.

```
//question1
//1
int lecture_points(float Points[N])
{
    int i=0;
    printf("entrer les Notes des etudiants au devoir de SR01  NB: Note entre [0 60]\n");
    for (i= 0; i < N; i++) {
        scanf("%4.2f",&Points[i]);
    }
    return 0;
}
```

2. Les trois fonctions qui suivent permettent de rechercher et d'afficher les notes maximale, minimale et moyenne de SR01
  - La note maximale du devoir SR01,

```
//question2
//2
int afficher_note_max(float Points[N])
{
    float NoteMax = 0;
    int i = 0;
    for(i=0; i<N; i++)
    {
        if(Points[i] > NoteMax) NoteMax = Points[i];
    }
    printf("La note maximale du devoir de SR01 est %f\n",NoteMax);
    return 0;
}
```

- La note minimale du devoir SR01,

```
//3
int afficher_note_min(float Points[N])
{
    float NoteMin = 60;
    int i = 0;
    for(i=0; i<N; i++)
    {
        if(Points[i] < NoteMin) NoteMin = Points[i];
    }
    printf("La note minimale du devoir de SR01 est %f\n",NoteMin);
    return 0;
}
```

- La moyenne des notes du devoir SR01.

```
//4
int moyenne_notes(float Points[N])
{
    float SommeNotes = 0;
    int i = 0;
    for(i=0; i<N; i++)
    {
        SommeNotes += Points[i];
    }
    printf("La moyenne des notes du devoir de SR01 est %f\n",SommeNotes/N);
    return 0;
}
```

3. Ce programme permet d'établir un tableau NOTES de dimension 7 à partir des POINTS des étudiants, qui est composé de la façon suivante :

- NOTES[6] contient le nombre de notes 60
- NOTES[5] contient le nombre de notes de 50 à 59
- NOTES[4] contient le nombre de notes de 40 à 49
- ...
- NOTES[0] contient le nombre de notes de 0 à 9

```
//5
//question3
int remplir_tableau_notes(float Points[N],int Notes[7])
{
    int i=0;
    //on va faire un controle sur la valeur de Points[i] pour savoir dans quelle intervalle se situe avant
    //d'incrémenter Notes[i]
    for(i=0; i<N; i++)
    {
        if(Points[i] == 60) Notes[6]++;
        if(Points[i]>=50 && Points[i]<=59) Notes[5]++;
        if(Points[i]>=40 && Points[i]<=49) Notes[4]++;
        if(Points[i]>=30 && Points[i]<=39) Notes[3]++;
        if(Points[i]>=20 && Points[i]<=29) Notes[2]++;
        if(Points[i]>=10 && Points[i]<=19) Notes[1]++;
        if(Points[i]>=0 && Points[i]<=9) Notes[0]++;
    }
    return 0;
}
```



4. Ce programme établit un graphique en nuage de points représentant le tableau notes. Le symbole 'o' est utilisé pour représenter le point dans le graphique et le domaine des notes est affiché en dessous du graphique.

```
//question4
int affiche_notes_rond(int Notes[7])
{
    int MaxN = 0;
    int position=0;// pour garder la position dela dernière case de valeur Notes[i]
    int i=0;
    int j=0;
    int k=0;
    //int Notes[7]={7,8,4,9,7,0,7};

    //je recherche le nombre max d'etudiant ayant eu une note dans un intervalle
    for(i=0; i<7;i++) if(Notes[i] > MaxN) MaxN = Notes[i];

    printf("\n");

    for(i=0; i<MaxN; i++) {

        printf("\n %d >",MaxN-i);

        for(j=0; j<7; j++){

            if(Notes[j] == MaxN-i) {

                if(MaxN-i){
                    //on affiche les rond en tenant compte du fait que il pourrais avoir des identiques
                    if(Notes[j]==Notes[position]){

                        //printf("\t");
                        for(k=0;k<(6*(j-position) + 3); k++) printf(" ");

                    } else

                        for(k=0;k<(6*j + 3); k++) printf(" ");

                    printf("o");

                }
                position = j;

            }

        }

        printf("\n");

    }
}
```

```

printf("\n    ");
//on affiche l'abscisse
for(j=0; j<7;j++){
    printf("+---");
    if(Notes[j]==0) {
        printf("o--");
    }else
        printf("---");
    }
    printf("+ \n    ");

    //on affiche les intervalles
    printf("| 0-9 | 10-19| 20-29| 30-39| 40-49| 50-59| 60 |");
    return 0;
}

```

5. Ce programme établir un graphique en bâtons représentant le tableau NOTES.

```

//7
//question5

int affiche_notes_baton(int Notes[7])
{
    int MaxN = 0;
    int position=0;// pour garder la position dela dernière case de valeur Notes[i]
    int i=0;
    int j=0;
    int k=0;
    // int Notes[7]={7,8,4,9,7,0,7};

    //je recherche le nombre max d'etudiant ayant eu une note dans un intervalle
    for(i=0; i<7;i++) if(Notes[i] > MaxN) MaxN = Notes[i];

    printf("\n");

    for(i=0; i<MaxN; i++) {

        printf("\n %d >",MaxN-i);

        for(j=0; j<7; j++){

```

```

        if(Notes[j] == MaxN-i) {
            if(MaxN-i){
//on affiche les rond en tenant compte du fait que il pourrais avoir des identiques
                if(Notes[j]==Notes[position]){
                    // printf(" ");
                    for(k=0;k<(6*(j-position)); k++) printf(" ");

                } else
                    for(k=0;k<(6*j); k++) printf(" ");

                printf(" ##### ");
            }
            position = j;
        }
    }
}

```

```

        printf("\n");
    }

    printf("\n    ");
    //on affiche l'abscisse
    for(j=0; j<7;j++){
        printf("+---");
        if(Notes[j]==0) {
            printf("o--");
        }else
            printf("---");
        }
        printf("+ \n    ");

    //on affiche les intervalles
    printf("| 0-9 | 10-19| 20-29| 30-39| 40-49| 50-59| 60 |");
    return 0;
}

```

### EXERCICE 3

L'objectif de cet exercice est de collecter un ensemble de données relatives à des restaurants et de les manipuler par la suite. Les informations concernant chaque restaurant sont stockées dans un fichier (restau.txt).

Une structure Restaurant est tout d'abord déclarer possédant les champs `nom_restaurant`, `adresse_restaurant`, `position_restaurant` et sa spécialité

```
/*#####Déclaration de la structure Restaurant#####*/
typedef struct rest {
    char nom_restaurant[STR_MAX_LENGTH];
    char adresse_restaurant[STR_MAX_LENGTH];
    double position_restaurant[2];
    char specialite[STR_MAX_LENGTH];
} Restaurant;
```

Le programme est constitué de 6 fonctions qui sont :

1. La fonction `int lire_restaurant (char* chemin, Restaurant restaurants [])`, qui récupère dans le tableau `restaurants` toutes les informations concernant chaque restaurant inséré dans le fichier nommé `chemin`. La fonction `lire_restaurant` retournera en fin de lecture le nombre de restaurants lu à partir du fichier.

Code:

```
// Fonction pour lire les restaurants depuis le fichier
int lire_restaurant(char *chemin, Restaurant restaurants[]) {
    FILE *file = fopen("restau.txt", "r");
    if (file == NULL) {
        printf("Error opening file %s\n", chemin);
        return -1; // Return -1 to indicate an error
    }

    char line[STR_MAX_LENGTH];
    int restaurantCount = 0;

    while (fgets(line, sizeof(line), file) != NULL &&
           restaurantCount < RESTAURANTS_NBR) {
        // Skip the header line or lines not following the expected format
        if (strstr(line, "Restaurant; adresse; coordonnee; specialite") != NULL) {
            continue;
        }

        char *token = strtok(line, ";");
        if (token != NULL) {
            strcpy(restaurants[restaurantCount].nom_restaurant, token);

            token = strtok(NULL, ";");
            if (token != NULL) {
                strcpy(restaurants[restaurantCount].adresse_restaurant, token);

                token = strtok(NULL, ";");
                if (token != NULL) {
                    // Extracting x and y coordinates
                    sscanf(token, "(x=%lf, y=%lf)",
                        &restaurants[restaurantCount].position_restaurant[0],
                        &restaurants[restaurantCount].position_restaurant[1]);

                    token = strtok(NULL, ";");
                    if (token != NULL) {
                        // Removing the leading '{' and trailing '}' from specialite
                        // Removing the leading '{' and trailing '}' from specialite
                        memmove(token, token + 1, strlen(token));
                        token[strlen(token) - 2] = '\0';
                        strcpy(restaurants[restaurantCount].specialite, token);

                        restaurantCount++;
                    }
                }
            }
        }
        fclose(file);
        return restaurantCount;
    }
}
```

Résultat:

```
Menu:
1. Chercher des restaurants dans un rayon
2. Chercher des restaurants par spécialité
3. Ajouter un restaurant
4. Afficher le nombre des restaurants
5. Quitter
Votre choix : 4
Le nombre de restaurants égale à : 21
```

2. La fonction `inserer_restaurant(Restaurant restaurant)` permettant d'insérer un restaurant à la fin du fichier (restau.txt).

Code:

```
// Fonction pour insérer un restaurant à la fin du fichier
void inserer_restaurant(Restaurant restaurant) {
    FILE *fichier = fopen("restau.txt", "r+");
    if (fichier == NULL) {
        printf("Erreur lors de l'ouverture du fichier.\n");
        return;
    }
    fseek(fichier, 0, SEEK_END);
    fprintf(fichier, "\n%s; %s; (x=%f, y=%f); {%s}", restaurant.nom_restaurant,
        restaurant.adresse_restaurant, restaurant.position_restaurant[0],
        restaurant.position_restaurant[1], restaurant.specialite);
    fclose(fichier);
}
```

Résultat :

```
Menu:
1. Chercher des restaurants dans un rayon
2. Chercher des restaurants par spécialité
3. Ajouter un restaurant
4. Afficher le nombre des restaurants
5. Quitter
Votre choix : 3
Entrez le nom du restaurant : Burger
Entrez l'adresse du restaurant : Compiègne
Entrez la position du restaurant (x y) : 12 15
Entrez les spécialités du restaurant : viande,poisson
```

```
Le Timgad; 21, rue Brunel - Paris 17eme;(x=75.89, y=78.52); {Cuisine marocaine};

La Famiglia; 2, rue Waldeck-Rousseau - Paris 17eme;(x=42.5, y=24.19); {Cuisine italienne};

Unico; 15, rue Paul Bert - Paris 11eme;(x=17.8, y=60.18); {Viande, rotisserie};
Burger; Compiègne; (x=12.000000, y=15.000000); {viande,poisson}
```

- La fonction `void cherche_restaurant(double x, double y, double rayon_recherche, Restaurant results[])` qui prend en paramètre la position actuelle d'un utilisateur et un rayon de recherche et qui remplit le tableau `results` par les restaurants présents dans le rayon de recherche.

Code:

```
// Fonction pour chercher les restaurants dans un rayon de recherche
void recherche_restaurant(double x, double y, double rayon_recherche,
                          const Restaurant restaurants[],
                          Restaurant results[]) {
    int resultsIndice = 0;
    for (int i = 0; i < RESTAURANTS_NBR; i++) {
        double distance_restaurant =
            distance(x, y, restaurants[i].position_restaurant[0],
                   restaurants[i].position_restaurant[1]);
        if (distance_restaurant <= rayon_recherche) {
            results[resultsIndice] = restaurants[i];
            resultsIndice++;
        }
    }
}
```

Résultat :

```
Menu:
1. Chercher des restaurants dans un rayon
2. Chercher des restaurants par spécialité
3. Ajouter un restaurant
4. Afficher le nombre des restaurants
5. Quitter
Votre choix : 1
Entrez votre position (x y) et le rayon de recherche : 12 15 20
Résultats de la recherche :
41 Penthievre, 41 Penthievre 41, rue de Penthievre - Paris 8eme, (0.000000, 0.000000), {Cuisine t
6 New York, 6, avenue de New York - Paris 16eme, (0.000000, 0.000000), {Cuisine gastronomiqu
African Lounge, 20 bis, rue Jean Giraudoux - Paris 16eme, (0.000000, 0.000000), {Cuisine africain
Le 122, 122, rue de Grenelle - Paris 7eme, (0.000000, 0.000000), {Cuisine gastronomiq
Le 41 Pasteur, 41, boulevard Pasteur - Paris 15eme, (0.000000, 0.000000), {Bistro
Belle Armee, 3, avenue de la Grande Armee - Paris 16eme, (2.500000, 12.578000), {Brasseri
Un Air de Famille, 118, rue des Dames - Paris 17eme, (0.000000, 0.000000), {Bistro
Le Bistro Mavrommatis Passy, 70, avenue Paul Doumer - Paris 16eme, (10.500000, 10.800000), {Cuisine grec
Al Dente, 38, rue de Varenne - Paris 7eme, (0.000000, 0.000000), {Cuisine italienn
```



4. La fonction `void cherche_par_specialite(double x, double y, char * specialite[], Restaurant results[])` qui prend en paramètre la position actuelle d'un utilisateur et une liste de spécialité et qui remplit le tableau `results` par les restaurants satisfaisants au moins une des spécialités passées en paramètre.

```
// Fonction pour chercher les restaurants par spécialité
void cherche_par_specialite(double x, double y,
                           char specialite[][STR_MAX_LENGTH],
                           int nbr_specialite, Restaurant restaurants[],
                           Restaurant results[]) {
    int resultsIndice = 0;

    for (int i = 0; i < RESTAURANTS_NBR; i++) {
        for (int j = 0; j < nbr_specialite; j++) {
            if (strcmp(restaurants[i].specialite, specialite[j])) {
                results[resultsIndice] = restaurants[i];
                resultsIndice++;
                break;
            }
        }
    }

    for (int i = 0; i < resultsIndice; i++) {
        for (int j = i + 1; resultsIndice; j++) {
            int x_1 = results[i].position_restaurant[0];
            int y_1 = results[i].position_restaurant[1];
            int x_2 = results[j].position_restaurant[0];
            int y_2 = results[j].position_restaurant[1];
            if (distance(x, y, x_1, y_1) > distance(x, y, x_2, y_2)) {
                Restaurant temp = results[i];
                results[i] = results[j];
                results[j] = temp;
            }
        }
    }
}
```

5. La fonction outil *double distance(double x1, double y1, double x2, double y2)* permettant de calculer la distance entre deux points (utilisée pour trier les résultats).

```
// Fonction pour calculer la distance entre deux points (utilisée pour trier
// les résultats)
double distance(double x1, double y1, double x2, double y2) {
    return sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
}
```

6. La fonction main() permettant de proposer un menu à l'utilisateur permettant de faire appel aux différentes fonctions mentionnées ci-dessus.

```
int main() {
    Restaurant restaurants[RESTAURANTS_NBR];
    int nbRestaurants = lire_restaurant("restau.txt", restaurants);
    // printf("%d\n", nbRestaurants);

    // Menu principal pour l'utilisateur
    int choix;
    double x, y, rayon_recherche;
    char specialites[MAX_SPECIALITES][STR_MAX_LENGTH];
    int nbSpecialites = 0;
    Restaurant nouveauRestaurant;

    do {
        printf("Menu:\n");
        printf("1. Chercher des restaurants dans un rayon\n");
        printf("2. Chercher des restaurants par spécialité\n");
        printf("3. Ajouter un restaurant\n");
        printf("4. Afficher le nombre des restaurants\n");
        printf("5. Quitter\n");
        printf("Votre choix : ");
        scanf("%d", &choix);

        switch (choix) {
            case 1:
                printf("Entrez votre position (x y) et le rayon de recherche : ");
                scanf("%lf %lf %lf", &x, &y, &rayon_recherche);
                Restaurant resultatsRayon[RESTAURANTS_NBR];
                recherche_restaurant(x, y, rayon_recherche, restaurants, resultatsRayon);
                printf("Résultats de la recherche :\n");
                for (int i = 0;
                    i < RESTAURANTS_NBR && resultatsRayon[i].nom_restaurant[0] != '\0';
                    i++) {
                    printf("%s, %s, (%f, %f), %s\n", resultatsRayon[i].nom_restaurant,
                        resultatsRayon[i].adresse_restaurant,
                        resultatsRayon[i].position_restaurant[0],
```

```
        resultatsRayon[i].position_restaurant[0],
        resultatsRayon[i].position_restaurant[1],
        resultatsRayon[i].specialite);
    }
    break;

case 2:
    printf("Entrez votre position (x y) : ");
    scanf("%lf %lf", &x, &y);
    printf(
        "Entrez les spécialités recherchées (séparées par des virgules) : ");
    char specialitesStr[STR_MAX_LENGTH];
    // fgets(specialitesStr, sizeof(specialitesStr), stdin);
    scanf("%s", specialitesStr);
    /*char *spe = strtok(specialitesStr, ",");
    nbSpecialites = 0;
    while (spe != NULL && nbSpecialites < MAX_SPECIALITES) {
        strcpy(specialites[nbSpecialites], spe);
        spe = strtok(NULL, ",");
        nbSpecialites++;
    }*/
    Restaurant resultatsSpecialite[RESTAURANTS_NBR];
    cherche_par_specialite(x, y, specialites, nbSpecialites, restaurants,
        resultatsSpecialite);
    printf("Résultats de la recherche :\n");
    for (int i = 0; i < RESTAURANTS_NBR &&
        resultatsSpecialite[i].nom_restaurant[0] != '\0';
        i++) {
        printf("%s, %s, (%f, %f), %s\n", resultatsSpecialite[i].nom_restaurant,
            resultatsSpecialite[i].adresse_restaurant,
            resultatsSpecialite[i].position_restaurant[0],
            resultatsSpecialite[i].position_restaurant[1],
            resultatsSpecialite[i].specialite);
    }
```

```
case 3:
    // Restaurant nouveauRestaurant;
    printf("Entrez le nom du restaurant : ");
    // fgets(nouveauRestaurant.nom_restaurant,sizeof(nouveauRestaurant.nom_restaurant),
    // stdin);
    scanf("%s", nouveauRestaurant.nom_restaurant);
    printf("Entrez l'adresse du restaurant : ");
    // fgets(nouveauRestaurant.adresse_restaurant,sizeof(nouveauRestaurant.adresse_restaurant),stdin);
    scanf("%s", nouveauRestaurant.adresse_restaurant);
    printf("Entrez la position du restaurant (x y) : ");
    scanf("%lf %lf", &nouveauRestaurant.position_restaurant[0],
    &nouveauRestaurant.position_restaurant[1]);
    printf("Entrez les spécialités du restaurant : ");
    // fgets(nouveauRestaurant.specialite,
    // sizeof(nouveauRestaurant.specialite),stdin);
    scanf("%s", nouveauRestaurant.specialite);
    inserer_restaurant(nouveauRestaurant);
    break;

case 4:
    if (nbRestaurants >= 0) {
        printf("Le nombre de restaurants égale à : %d\n", nbRestaurants);
    } else {
        printf("Erreur lors de l'ouverture du fichier.\n");
    }
    break;

case 5:
    printf("Au revoir !\n");
    break;

default:
    printf("Choix invalide. Veuillez réessayer.\n");
}
} while (choix != 5);

return 0;
```