

Devoir 1 SR01

A remettre pour 25 octobre 2023

- Ce devoir peut se faire en binôme. Les deux étudiants formant un binôme doivent appartenir au même groupe de TD.
- Les différents binômes devront remettre un seul fichier ***.zip** contenant les codes sources ainsi qu'un rapport électronique (pdf) de quelques pages.
- Le dernier délai pour la remise est fixé pour le **25 octobre 2023**.

1 Exercice 1

Pour chacun de ces programmes, exécuter le programme et donner une explication du résultat obtenu:

```
1 #include <stdio.h>
2 int main() {
3     int A=20, B=5;
4     int C=!--A/++!B;
5     printf("A=%d B=%d c=%d \n", A, B, C);
6 }
```

Listing 1: programme 1

```
1 #include <stdio.h>
2 int main() {
3     int A=20, B=5, C=-10, D=2;
4     printf("%d \n", A&&B || !0&&C++&&!D++);
5     printf("c=%d d=%d \n", C, D);
6 }
```

Listing 2: programme 2

```
1 #include <stdio.h>
2 int main() {
3     int p[4]={1, -2, 3, 4};
4     int *q=p;
5     printf("c=%d\n", *++q**q++);
6     printf("c=%d \n", *q);
7 }
```

Listing 3: programme 3

```
1 #include <stdio.h>
2 int main() {
3     int p[4]={1, -2, 3, 4};
4     int *q=p;
5     int d=*q&*q++|*q++;
6     printf("d=%d\n", d);
7     printf("q=%d \n", *q);
8 }
```

Listing 4: programme 4

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

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

Listing 5: programme 5

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

Listing 6: programme 6

2 Exercice 2

1. Écrire un programme qui lit 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 .
2. Écrire des programmes pour rechercher et afficher :
 - La note maximale du devoir SR01,
 - La note minimale du devoir SR01,
 - La moyenne des notes du devoir SR01.
3. A partir des POINTS des étudiants, établir un tableau NOTES de dimension 7 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
4. Établir un graphique en nuage de points représentant le tableau notes. Utilisez le symbole 'o' pour représenter le point dans le graphique et affichez le domaine des notes en dessous du graphique.
5. Établir un graphique en bâtons représentant le tableau NOTES. Utilisez les symboles ou autres pour la représentation des bâtons et affichez le domaine des notes en dessous du graphique.

[illegible]

1000

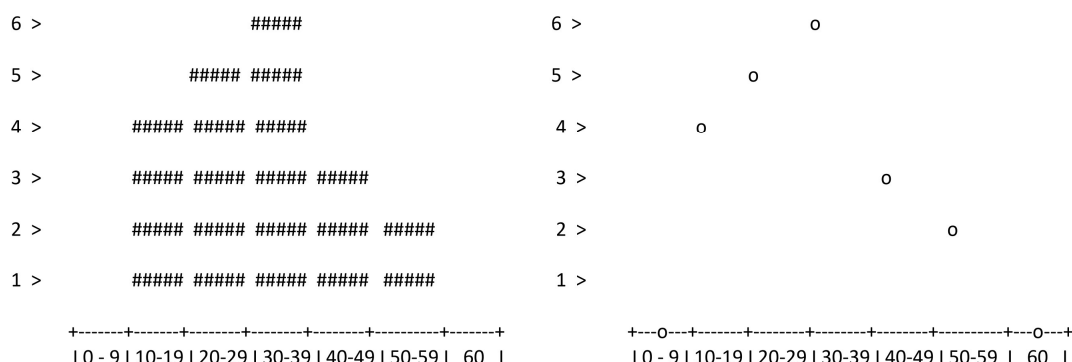


Figure 1: Exemple d’affichage.

3 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) selon le format donné dans la figure 2.

Dans ce programme vous êtes sensés :

1. Déclarer une structure `Restaurant` contenant les champs suivants : `nom_restaurant`, `adresse_restaurant`, `position_restaurant` et sa spécialité.
2. A travers la fonction `int lire_restaurant(char* chemin, Restaurant restaurants [])`, récupérer 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.
3. Implémenter la fonction `inserer_restaurant(Restaurant restaurant)` permettant d'insérer un restaurant à la fin du fichier (`restau.txt`) .
4. Implémenter 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.

```
Restaurant; adresse; coordonnée; spécialité

Edern; 6, rue Arsène Houssaye - Paris 8ème;(x=1.5, y=44.8); {Cuisine gastronomique};
La Farnesina; 9, rue Boissy d'Anglas - Paris 8ème;(x=70.5, y=74.12578); {Cuisine italienne};
41 Penthievre; 41 Penthievre 41, rue de Penthievre - Paris 8ème; (x=40.5, y=77.12578); {Cuisine traditionnelle française};
6 New York; 6, avenue de New York - Paris 16ème; (x=12.5, y=74.12578); {Cuisine gastronomique};
African Lounge; 20 bis, rue Jean Giraudoux - Paris 16ème; (x=18.5, y=7.578); {Cuisine africaine};
Le 122; 122, rue de Grenelle - Paris 7ème; (x=48.8571, y=2.319250000000011); {Cuisine gastronomique};
Le 404; 69, rue des Gravilliers - Paris 3ème;(x=48.8645, y=2.3542); {Cuisine marocaine};
Le 41 Pasteur; 41, boulevard Pasteur - Paris 15ème; (x=48.8426, y=2.3132699999999886); {Bistrot};
Belle Armée; 3, avenue de la Grande Armée - Paris 16ème;(x=2.5, y=12.578); {Brasserie};
Benkay; 61, quai de Grenelle - Paris 15ème;(x=142.5, y=7.1); {Cuisine japonaise};
Agapé; 51, rue Jouffroy-d'Abbans - Paris 17ème;(x=128.5, y=200.178); {Haute cuisine française};
```

Figure 2: Exemple d'un fichier de données

5. Implémenter 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.

Remarque : pour la dernière question, les restaurants insérés dans le tableau *results* doivent être ordonnés du plus proche au plus loin par rapport à la position passée en paramètre.

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