

Compte-rendu du projet C

« It's time to C Code (2) ! »

Membres du groupe :

- Anto BENEDETTI ;
- Antony DAVID ;
- Anthony JABRE.

Sommaire

Durant ce projet de programmation en langage C, nous avons cherché, pour chaque exercice :

- À avoir le code le plus lisible possible ;
- À ajouter des détails rendant l'expérience utilisateur plus agréable ;
- À sécuriser au maximum le programme, en prenant en compte le maximum de cas différents, lors de la saisie par l'utilisateur.

Avant de commencer l'explication de chaque exercice, nous devons décrire trois fonctionnalités communes à tous :

❖ Boucle principale

Quasiment l'ensemble du code est à l'intérieur d'une boucle « *do while* », où la condition n'est d'autre qu'une variable « *running* » valant soit 1, soit 0.

Elle représente donc un booléen, la rendant exploitable par la boucle. Sa valeur est initialisée à 1 au tout début du code, et est mise à 0 uniquement lorsque l'utilisateur choisi d'arrêter le programme, grâce au menu.

❖ Menu

Dans chaque programme, un menu est accessible pour lancer l'exercice, autant de fois que l'on veut, ainsi que pour l'arrêter.

Cette fonctionnalité nécessite, dans un premier temps, à afficher le menu, grâce à un « *printf()* », et à récupérer le choix de l'utilisateur, puis dans un second temps, à utiliser un « *switch* ».

Le premier choix correspond à lancer le programme (case 1 dans le switch), contenant l'algorithme de l'exercice. Le second choix correspond à l'arrêt du programme.

❖ Vérification des saisies

Chaque saisie par l'utilisateur est contrôlée par un algorithme en trois vérifications :

- Type de la variable respectée ;
- Inférieur au minimum ;
- Supérieur au maximum

Lorsqu'une étape est validée, un compteur « *check_step* » s'incrmente. Cette variable doit être égale à 3 pour que la saisie soit acceptée. Si la vérification échoue, le compteur est réinitialisé et la boucle « *while* » recommence.

Une variable « *type_check* » est utilisée pour vérifier si l'utilisateur a bien rentré une valeur correspondant au type de variable demandé.

Si le programme nécessite une valeur numérique entière (int par exemple), mais que l'utilisateur entre une chaîne de caractères, la variable sera égale à 0.

Dans le cas où le type est respecté, la variable sera égale aux nombres de types respectés.

Exemple :

```
type_check = scanf("%d %c", &a, &b);
```

- ➔ Si la première saisie de l'utilisateur est un nombre entier, et que la deuxième est un caractère, « *scanf()* » retourne la valeur 2, qui sera affectée à la variable « *type_check* ».
- ➔ Si la première saisie est un nombre entier, mais que la deuxième est autre qu'un caractère, « *scanf()* » retourne la valeur 1, qui sera affectée à la variable « *type_check* ».
- ➔ Si aucun des saisies ne correspond au code format, « *scanf()* » retourne la valeur 0, qui sera affectée à la variable « *type_check* ».

Après le contrôle réussi, la variable « *check_step* » est de nouveau réinitialisée, afin que la prochaine vérification ne soit pas ignorée.

Exercices

❖ Exercice 1