

Distributeur automatique de boissons

1 Mise en situation

Vous êtes chargé de développer un logiciel permettant le fonctionnement d'un distributeur de boissons. La partie opérative sera remplacée par des interactions avec l'utilisateur à l'aide de son clavier et de son écran.



2 Fonctionnement de l'application

L'utilisateur voit apparaître un menu lui permettant de choisir une boisson. Lorsque ce choix est effectué, le prix et le nom de la boisson s'affichent, un nouveau menu apparaît pour la saisie des pièces à introduire. Au fur et à mesure du paiement le montant à payer diminue, jusqu'à atteindre le prix de la boisson ou une somme supérieure dans ce cas si cela est possible, le distributeur redonne la monnaie ou demande de faire l'appoint. La boisson distribuée, son montant réglé, le distributeur est prêt à proposer une autre boisson.

3 Mise en oeuvre

3.1 Création du projet :

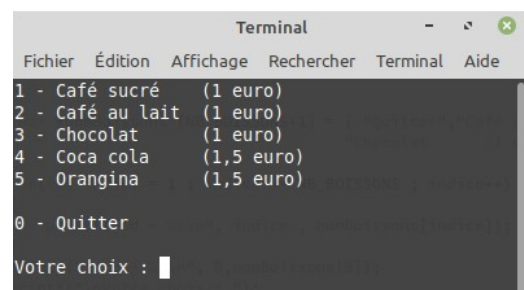
Avec NetBeans, créez une application en mode console utilisant le langage C sous la forme d'un projet nommé « **Distributeur** ». Il sera composé d'un fichier **main.c** contenant le programme principal, d'un fichier **menu.c** contenant le corps des fonctions pour l'affichage qui vont être développées dans cette application et un fichier **menu.h** contenant les prototypes de ces fonctions.

3.2 Affichage du Menu des Boissons : **Étape 1**

La fonction **AfficherMenu** permet la création d'un menu tel l'exemple ci-dessous. Cette fonction reçoit en paramètre d'entrée le tableau contenant le nom des différentes boissons et le nombre de boissons. Elle retourne le numéro choisi sous la forme d'un entier ou 0 si l'utilisateur désire quitter le programme. La fonction efface l'écran après chaque appui sur une touche et n'accepte que les touches indiquées dans le menu.

Pour effacer l'écran vous utiliserez la fonction `system` de la librairie `stdlib.h` en passant en paramètre la chaîne correspondant à la commande `bash clear`.

```
system("clear");
```



```
Terminal
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
1 - Café sucré      (1 euro)
2 - Café au lait   (1 euro)
3 - Chocolat       (1 euro)
4 - Coca cola      (1,5 euro)
5 - Orangina       (1,5 euro)
0 - Quitter
Votre choix : █
```

Avant le programme principal, définissez la constante **NB_BOISSON** initialisée avec la valeur 5. Dans la fonction **main**, déclarez le tableau **nomBoisson** contenant des chaînes de caractères de la manière suivante :

Déclaration du tableau de boissons

main.c

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

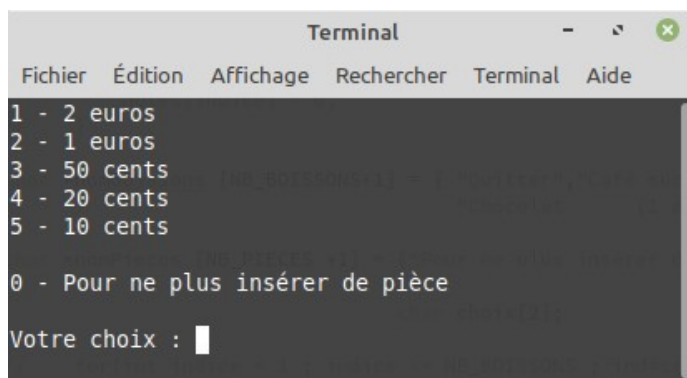
#define NB_BOISSONS 5

int main()
{
    char *nomBoissons [NB_BOISSONS +1] = { "Quitter", "Café sucré (1 euro)",
                                              "Café au lait (1 euro)", "Chocolat (1 euro)",
                                              "Coca cola (1,5 euro)", "Orangina (1,5 euro)"};
}
```

Complétez le programme principal pour qu'il affiche le menu des boissons, affiche la boisson choisie, demande d'appuyer sur une touche, efface l'écran et affiche à nouveau le menu jusqu'à l'appui sur la touche 0.

3.3 Gestion de l'introduction des pièces : Étape 2

Complétez la fonction **main** de telle sorte qu'un menu apparaisse pour choisir les pièces introduites comme le montre la figure ci-dessous :



Ajouter à votre projet les fichiers **monnayeur.c** et **monnayeur.h** pour recevoir les fonctions liées au monnayeur du distributeur de boissons

Ajoutez à la fonction **main** un tableau à 2 dimension **piecesIntroduites** contenant d'une part la valeur de chaque pièce exprimé en cents et d'autre part le nombre de chaque type de pièce introduite dans l'appareil.

Sa déclaration : `int piecesIntroduites[2][NB_PIECES] ;`

La constante **NB_PIECES** indiquant le nombre de type de pièce admise par le monnayeur du distributeur est à déclarée dans le fichier **monnayeur.h**.

Réaliser la fonction **CalculerMontantDepose** dont le rôle est de retourner la somme du nombre de pièces multiplié par son montant. Le prototype de cette fonction est le suivant : `int CalculerMontantDepose(int _pieces[][NB_PIECES], int _nbPieces) ;`

Ajoutez à la fonction **main** un tableau **prixBoissons** contenant le prix de chaque boisson. Attention, le prix indiqué à l'indice 0 doit être égal à 0 (correspond à quitter dans le menu).

Complétez à nouveau la fonction **main** pour que l'utilisateur puisse insérer ces pièces dans le distributeur jusqu'à l'appui sur 0 ou qu'il est atteint le prix de la boisson. Au fur et à mesure le montant introduit s'affiche sur l'écran.

3.4 Restitution de la monnaie : Étape 3

L'étude se décompose en deux parties :

- La première permet de rendre la monnaie par rapport à un type de pièce, c'est l'objet de l'algorithme **RendrePiece**. Réalisez cet algorithme.
- La seconde permet de rendre complètement la monnaie avec tous les types de pièces. Ecrire l'algorithme **RendreMonnaie**.

Pour ces deux algorithmes réfléchissez aux paramètres nécessaires

Complétez la fiche de test permettant le rendu de la monnaie en complétant le tableau fourni en annexe.

Lorsque les algorithmes sont satisfaisant procéder au codage.

Ajoutez à la fonction **main** un tableau **stockPieces** à 2 dimensions (1 pour la valeur des pièces en cents et l'autre pour le nombre de pièce en stock) initialisé pour n'avoir aucune pièce de 2€ et de 1€, avoir 5 pièces de 50 cents, 4 pièces de 20 et 2 pièces de 10.

Vous ajoutez également un tableau **monnaie** également à 2 dimensions qui contiendra le nombre de pièces à rendre pour chaque sorte de pièces.

Compléter le programme principal pour rendre toutes les sortes de pièces. Dans un deuxième temps, affichez le nombre de pièces rendu dans chaque catégorie, si ce nombre est différent de 0.

3.5 Test unitaire : Sous la forme d'un compte-rendu de test

Procédez au test unitaire de rendu de monnaie et complétez le compte-rendu de test en utilisant le tableau suivant le tableau suivant :

Compte-rendu du Test F01	Date :	Version :
Nom du testeur :		
Opération x :	Le résultat obtenu n'est pas conforme aux attentes (description), la solution au problème est ...	

Nom :	Prénom :	Snir 1	2021-2022
-------	----------	--------	-----------

Jeu de test à compléter :

[illegible]