

DS N° 1 - INFORMATIQUE I

Calculatrice et documents non autorisés

Lorsqu'aucune consigne n'est précisée, les réponses peuvent être données en langage C ou en pseudo-code.

Exercice 1 (*Distributeur de billets*)

Dans cet exercice, nous allons créer un programme simulant le fonctionnement d'un distributeur de billets.

Pour simplifier le problème nous allons considérer que le client qui utilise le distributeur a un solde de 2500 euros et un code de carte bleue de 4242.

Voici les règles de fonctionnement du distributeur :

- **Authentification :**
 - Le client a deux essais pour entrer le code de sa carte bleue.
 - Si le code est incorrect après deux essais, le programme affiche un message d'erreur et s'arrête.
- **Opération de Retrait :**
 - Si le code est correct, le client entre le montant qu'il souhaite retirer.
 - Le programme vérifie si le montant demandé n'est pas négatif ou nul ou bien s'il ne dépasse pas le solde du client.
 - Si le montant n'est pas correct, le programme affiche un message d'erreur et l'opération est annulée.
- **Distribution des Billets :**
 - Le distributeur dispose de billets de 200, 50, 20 et 5 euros.
 - Si le montant entrée précédemment est correct, le programme calcule la répartition des billets en fonction du montant demandé.
 - Les billets sont distribués en priorité par ordre décroissant (200, puis 50, puis 20, puis 5).
- **Affichage des Résultats :**
 - Si une opération est annulée (code incorrect, solde insuffisant), le programme affiche le message approprié comme précisé plus haut.
 - Sinon le programme affiche la quantité de billets distribués pour chacune des catégories de billets, et affiche également le solde restant sur le compte du client.

Exercice 2 (*Multiplexeur*)

Un multiplexeur est un circuit logique permettant de sélectionner une sortie parmi plusieurs entrées.

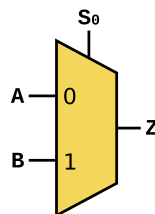


FIGURE 1 – Schéma d'un multiplexeur 2 vers 1

1. Dans l'exemple si dessus, la sortie Z vaut A si $S_0 = 0$, et vaut B si $S_0 = 1$. Déterminer l'équation de sortie du multiplexeur 2 vers 1 en fonction des deux entrées et du bit de sélection S_0 . Dessiner le circuit logique correspondant.
2. Comment faire pour sélectionner parmi 4 entrées plutôt que 2 ? Pour n entrées ? Argumenter.
3. Ecrire l'équation de sortie simplifiée permettant d'obtenir un multiplexeur 4 vers 1.
4. Comment obtenir un multiplexeur 4 vers 1 à partir de multiplexeurs 2 vers 1 ? Répondre à l'aide d'une argumentation précise et/ou d'un schéma : l'idéal étant de faire les deux.

Exercice 3 (*Intersection de deux intervalles entiers*)

Cet exercice doit être réalisé **en langage C obligatoirement**. Ne pas respecter cette consigne entrainera un zéro à l'exercice.

Ecrire un programme qui va :

- Demander à l'utilisateur de saisir deux valeurs entières a et b , pour définir le premier intervalle.
- Demander à l'utilisateur de saisir deux valeurs entières c et d , pour définir le deuxième intervalle.
- Déterminer et afficher l'intersection des deux intervalles entiers : $[a, b]$ et $[c, d]$.

Exemple :

- Si les intervalles sont $[5, 11]$ et $[3, 7]$, l'algorithme doit afficher l'intervalle résultant $[5, 7]$.
- Si les intervalles sont $[5, 19]$ et $[12, 20]$, l'algorithme doit afficher l'intervalle résultant $[12, 19]$.
- Si un intervalle est totalement inclu dans un autre, alors l'intervalle inclu est l'intervalle à afficher en sortie.

Ordre des bornes des intervalles :

- Le programme doit vérifier que les bornes de chaque intervalle sont dans le bon ordre (ordre croissant). Ex : $[12, 57]$ est valide, $[23, 9]$ ne l'est pas. Si l'un des intervalles a des bornes dans le mauvais ordre, le programme affiche un message d'erreur et s'arrête.

Cas d'Absence de Chevauchement :

- Dans les situations où les intervalles ne se chevauchent pas, par exemple $[8, 11]$ et $[1, 7]$, l'intersection doit être considérée comme vide. On affichera le message "Vide".

Cas d'Intersection à un Point :

- De plus, lorsque les intervalles partagent un seul point, comme dans $[1, 8]$ et $[8, 11]$, l'algorithme doit afficher l'intervalle $[8, 8]$.