

Exercice 1:

Une compagnie d'assurance automobile propose à ses clients quatre familles de tarifs identifiables par une couleur, du moins au plus onéreux: tarifs bleu, vert, orange et rouge. Le tarif dépend de la situation du conducteur:

- Un conducteur de moins de 25 ans et titulaire du permis depuis moins de deux ans, se voit attribuer le tarif rouge, si toutefois il n'a jamais été responsable d'accident. Sinon, la compagnie refuse de l'assurer.
- Un conducteur de moins de 25 ans et titulaire du permis depuis plus de deux ans, ou de plus de 25 ans mais titulaire du permis depuis moins de deux ans a le droit au tarif orange s'il n'a jamais provoqué d'accident, au tarif rouge pour un accident, sinon il est refusé.
- Un conducteur de plus de 25 ans titulaire du permis depuis plus de deux ans bénéficie du tarif vert s'il n'est à l'origine d'aucun accident et du tarif orange pour un accident, du tarif rouge pour deux accidents, et refusé au-delà.
- De plus, pour encourager la fidélité des clients acceptés, la compagnie propose un contrat de la couleur immédiatement la plus avantageuse s'il est entré dans la maison depuis plus d'un an.

Écrire l'algorithme puis le traduire en C++ permettant de saisir les données nécessaires (sans contrôle de saisie) et de traiter ce problème.

Algorithme d'assurance automobile

variables:

âge: entier

nombre année_du_permis: entier

Accident: chaîne de caractères(Oui/Nom)

Fidélité: chaîne de caractères(Oui/Nom)

Début

// Saisie des informations sur le conducteur

Afficher ("Veuillez saisir l'âge du conducteur : ")

Saisir (âge)

Afficher ("veuillez entrez le nombre d'années depuis l'obtention du permis :
")

saisir (nombre année du permis)

Afficher ("Avez-vous déjà provoqué un accident ? (Oui/Non) : ")

Saisir (accident)

// Vérification des conditions pour déterminer le tarif

Si (âge < 25) Alors

 Si (nombre d'année du permis < 2) Alors

 Si (accident = "Non")Alors

 Afficher ("Tarif : Rouge")

Sinon Afficher ("Refusé")
Fin Si

Sinon Si (accident = "Non")Alors
Afficher ("Tarif : Orange")
Sinon Afficher ("Tarif : Rouge")
Fin Si

Fin Si

Sinon Si (nombre d'année du permis < 2) Alors
Afficher "Refusé"

Sinon Si (accident = "Non") Alors
Afficher ("Tarif : Vert")
Sinon Si accident = "Un accident" Alors
Afficher "Tarif : Orange"
Sinon
Afficher ("Tarif : Rouge")
Fin Si

Fin Si

Fin Si

// Vérification de la fidélité du client

Afficher ("Le client est-il entré dans la maison depuis plus d'un an ?
(Oui/Non) : ")
Saisir (fidélité)
Si (fidélité = "Oui") Alors
Afficher ("Contrat de la couleur immédiatement la plus avantageuse.")
Fin Si

Finalgorithme

Traduction en C++

```
#include <iostream>
```

```
#include <string>
```

```
enum Tarif { Pas de Tarif, VERT, ORANGE, ROUGE, REFUSE };
```

```
Tarif determinerTarif(int age, int anneesPermis, int accidents, int anneesClient)
```

```
{
```

```
    if ((age < 25 && anneesPermis < 2 && accidents > 0) ||
```

```
        (accidents > 1) ||
```

```
        (age < 25 && anneesPermis >= 2 && accidents > 1) ||
```

```
        (age >= 25 && anneesPermis < 2 && accidents > 1)) {
```

```
        return REFUSE;
```

```
    }
```

```
    if (age < 25 && anneesPermis < 2) {
```

```
        return ROUGE;
```

```
    }
```

```
    if ((age < 25 && anneesPermis >= 2 && accidents == 1) ||
```

```
        (age >= 25 && anneesPermis < 2 && accidents == 1) ||
```

```
        (age >= 25 && anneesPermis >= 2 && accidents == 2)) {
```

```
        return ROUGE;
```

```
    }
```

```
    if ((age < 25 && anneesPermis >= 2 && accidents == 0) ||
```

```
        (age >= 25 && anneesPermis < 2 && accidents == 0) ||
```

```
        (age >= 25 && anneesPermis >= 2 && accidents == 1)) {
```

```
        return ORANGE;
```

```
    }
```

```
    if (age >= 25 && anneesPermis >= 2 && accidents == 0) {
```

```
        return VERT;
```

```
    }
```

```
    return Pas de Tarif;
```

```
}
```

```
int main() {
```

```
    int age, anneesPermis, accidents, anneesClient;
```

```
    std::cout << "Entrez l'age du conducteur: ";
```

```

std::cin >> age;
std::cout << "Entrez le nombre d'annees de permis: ";
std::cin >> anneesPermis;
std::cout << "Entrez le nombre d'accidents: ";
std::cin >> accidents;
std::cout << "Entrez le nombre d'annees en tant que client: ";
std::cin >> anneesClient;

```

```

Tarif tarif = determinerTarif(age, anneesPermis, accidents, anneesClient);

```

```

switch (tarif) {
    case VERT:
        std::cout << "Tarif Vert";
        break;
    case ORANGE:
        std::cout << "Tarif Orange";
        break;
    case ROUGE:
        std::cout << "Tarif Rouge";
        break;
    case REFUSE:
        std::cout << "Refuse";
        break;
}
return 0;
}

```

Exercice 2:

Écrire un algorithme qui permet de lire la suite des prix (en FCFA entiers et terminer par zéro) des achats d'un client. Le nombre total de produits achetés doit être inférieur à 10. Le prix des différents achats doit être mis dans un tableau.

Calculer:

La somme qu'il doit,

Lire la somme qu'il paye

Écrire un algorithme qui permet de lire la suite des prix (en FCFA entiers et terminer par zéro) des achats d'un client. Le nombre total de produits achetés doit être inférieur à 10. Le prix des différents achats doit être mis dans un tableau.

Calculer:

- La somme qu'il doit,
- Lire la somme qu'il paye et
- Simuler la remise de la monnaie en affichant les textes par exemple "3 billets de 1000 CFA", "1 billet de 500 CFA", "3 pièces de 100 FCFA", "1 pièce de 50 FCFA", "1 pièce de 25 FCFA", autant de fois qu'il y a de coupures de la sorte à rendre

Les coupures prises en compte sont les suivantes: 5000, 2000, 500, 200, 100, 50 et 25.

Algorithme des prix

Variables

total_produits = 0

prix_achats = []

prix_total = 0

somme_payée = entier

Monnaie = entier

coupures = [billet 5000, billet 2000, billet 1000, billet 500, pièce 200, pièce 100, pièce 50, pièce 25]entier

Début:

Tant que total_produits < 10 Faire

Afficher ("Entrez le prix (en CFA, et terminer par 0) : ")

lire (prix_produit)

Si prix_produits == 0 Alors

Arrêté

Fin Si

prix_total = prix_total + prix_produits

Afficher ("Les prix des produits achetés sont:", prix_achats)

total_achats = total_produits + 1

Fin Tant que

si (somme_payée < prix_total)

Afficher("le montant payé est insuffisant.")

Sinon si (Somme_payée = prix_total)

Afficher("paiement complète")

sinon

Monnaie = Somme_payée - Prix_total

Tant que (Monnaie >= 5000)

Afficher ("5 billets de 1000 CFA")

Monnaie = Monnaie - 5000

Fintantque

Tant que (Monnaie >= 2000 CFA)

Afficher ("1 billet de 2000 CFA")

Monnaie = Monnaie - 2000

Fintantque

Tant que (Monnaie >= 500)

Afficher ("1 billet de 500 CFA")

Monnaie = Monnaie - 500

Fintantque

Tant que (Monnaie >= 200)

Afficher ("1 pièce de 200 CFA")

Monnaie = Monnaie - 200

Fintantque

Tant que (Monnaie \geq 100)
 Afficher ("1 pièce de 100 CFA")
 Monnaie = Monnaie - 100
Fintantque

Tant que (Monnaie \geq 50)
 Afficher ("1 pièce de 50 CFA")
 Monnaie = Monnaie - 50
Fintantque

Tant que (Monnaie \geq 25)
 Afficher ("1 pièce de 25 CFA")
 Monnaie = Monnaie - 25
Fintantque

AFFICHER "Billets de 1000 FCFA : ", billet 1000
AFFICHER "Billets de 500 CFA : ", billet 500
AFFICHER "Pièces de 100 CFA : ", piece 100
AFFICHER "Pièces de 50 CFA : ", piece 50
AFFICHER "Pièces de 25 FCFA : ", pièce 25

Finalgorithme