

Table des matières

A Questions de cours	1
B Exercice	4

Ce sujet comprend 40 questions pour un total de 60 points.

A Questions de cours

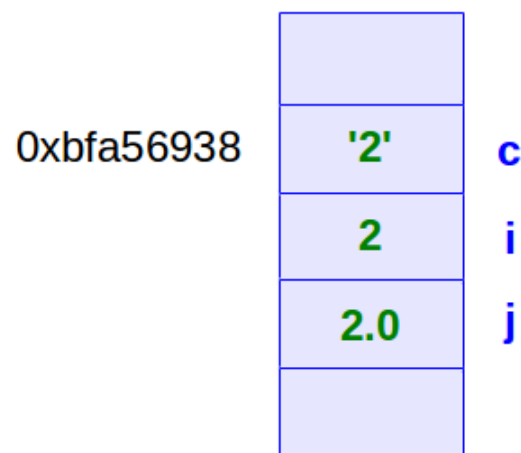
Question 1 (1 point)

Donner la définition d'une variable.

Question 2 (1 point)

Rappeler les règles de codage pour les variables.

Soit les cases mémoires suivantes :



Question 3 (1 point)

À quoi correspondent les termes `0xbfa56938`, `2` et `i` ?

Question 4 (1 point)

Donner les types des variables `c`, `i` et `j`.

Question 5 (1 point)

Est-ce que les cases mémoires des variables `c`, `i` et `j` contiennent la même valeur en bits ?

Question 6 (1 point)

Est-ce que les cases mémoires des variables `c`, `i` et `j` ont la même taille en octets ?

Question 7 (1 point)

Quelles sont les adresses mémoires des variables `c` et `j` ? On suppose que la variable `i` est de type *int*.

Question 8 (1 point)

Donner l'instruction qui affecte la valeur `3.5` à la variable `j` ? `j` sera la *lvalue* (cad un espace de stockage modifiable) et `3.14` une *rvalue*.

Question 9 (1 point)

Dans l'instruction précédente, quelle est la particularité de la valeur `3.14` ?

Question 10 (1 point)

Donner l'instruction qui affecte la valeur de j à la variable i ? i sera la *lvalue* et j une *rvalue*.

Question 11 (1 point)

Quelle sera alors la valeur stockée dans la variable i ?

Question 12 (1 point)

Quelle sera la valeur stockée dans la variable i si on exécute l'instruction suivante : $i = j * 0.5$?

Question 13 (1 point)

Quel est le problème dans cette instruction $3.5 = j$; ?

Question 14 (1 point)

Déclarer un pointeur p sur un entier.

Question 15 (1 point)

Affecter l'adresse de la variable i au pointeur p ?

Question 16 (1 point)

Affecter la valeur 5 à la variable i en utilisant le pointeur p ?

Question 17 (1 point)

Comment sont codés les nombres négatifs entiers?

Question 18 (1 point)

Quel serait le codage en binaire de la valeur entière -5 ?

Question 19 (1 point)

Qu'évoque pour vous la représentation *little-endian* et *big-endian* ?

Question 20 (1 point)

Donner la représentation binaire sur 32 bits de la valeur 5 en mémoire pour une architecture *Intel x86* ?

Question 21 (1 point)

Bonus : Est-ce que la représentation *little-endian* affecte le stockage d'une chaîne de caractères codée en ASCII ?

B Exercice

Remarque : considérer que les questions font partie d'un même programme C++.

Question 22 (3 points)

Définir une variable `reservoir`. Attention définir une variable sous entend qu'elle est aussi initialisée !

Question 23 (1 point)

Affecter 45.6 litres d'essence à votre variable `reservoir`.

Question 24 (2 points)

Définir une constante globale `PRIX_SUPER_95` égale à la valeur 1.48 (pour un 1 litre).

Question 25 (3 points)

Définir une fonction `calculePrix95()` qui reçoit le contenu en litre d'un réservoir et qui retourne le prix à payer pour cette quantité d'essence. On considère que le réservoir contient du Super95!

Question 26 (2 points)

En appelant la fonction `calculePrix95()`, donner l'instruction qui affichera le prix à payer pour ce réservoir. N'oubliez jamais de donner l'unité quand vous affichez une valeur à un utilisateur.

Question 27 (2 points)

Définir une fonction `calculeAutonomie()` qui reçoit le contenu en litre d'un réservoir et la consommation exprimée en litre pour 100 km. Elle retournera l'autonomie en km pour cette quantité d'essence sous forme entière.

Question 28 (1 point)

Donner l'instruction qui permettra d'afficher l'autonomie retournée par la fonction `calculeAutonomie` pour une consommation de 7.4 l.

Question 29 (1 point)

Définir une variable `consommation` qui contiendra la valeur de la consommation en litres pour 100 km. Attention définir une variable sous entend qu'elle est aussi initialisée!

Question 30 (1 point)

Définir une variable `autonomie` qui contiendra la valeur de l'autonomie en km (sous forme entière). Attention définir une variable sous entend qu'elle est aussi initialisée!

Question 31 (2 points)

Donner l'instruction qui permettra de calculer l'autonomie retournée par la fonction `calculeAutonomie` pour une consommation de 7.4 l.

Question 32 (3 points)

Définir une fonction `razReservoir()` qui vide le reservoir passé en argument et qui ne retourne rien. Attention, vous ne pouvez pas "passer par valeur" la variable `reservoir`!

Question 33 (1 point)

Afficher le contenu de la variable `reservoir` avant et après avoir appelé la fonction `razReservoir`.

Question 34 (3 points)

Définir une fonction `remplirReservoir()` qui remplit si nécessaire le reservoir pour parcourir une certaine distance et qui retourne la quantité en litres ajoutée à celui-ci. Le fonction recevra trois arguments : le reservoir à remplir, la consommation et la distance à parcourir en km sous forme entière.

Question 35 (4 points)

Définir une structure `Voiture` qui possèdera les membres suivants : le reservoir, la consommation et la distance parcourue en km sous forme entière.

Question 36 (2 points)

Allouer dynamiquement une variable `voiture` de type pointeur sur `Voiture`.

Question 37 (2 points)

Affecter les valeurs 99.9, 7.5 et 0 successivement aux membres `reservoir`, `consommation` et `distance` de la variable `voiture`.

Question 38 (2 points)

Libérer la mémoire allouée à la variable `voiture`.

Question 39 (2 points)

Déclarer un tableau `garage` de 10 éléments de type `Voiture`.

Question 40 (2 points)

Affecter les valeurs 0., 7.4, 100 successivement aux membres `reservoir`, `consommation` et `distance` de la quatrième `voiture` du `garage`.