TD n° 1

Révisions

Exercice 1.

Au musée de Marly, les tarifs d'entrée sont définis par les règles suivantes : gratuité pour les enfants d'au plus 6 ans, tarif réduit (3 euros) pour les enfants entre 7 et 12 ans et les personnes âgées à partir de 65 ans, et tarif plein (5 euros) pour les autres. De plus, les habitants de la ville de Marly ont droit à une réduction de 1 euro.

Pour diminuer le temps d'attente aux caisses, on souhaite programmer des machines en libre service pour l'achat du billet d'entrée.

1. Écrire une fonction **tarif** qui demande à l'utilisateur l'âge de la personne à laquelle le billet est destiné, puis qui demande si cette personne habite à Marly, et enfin qui affiche le prix à payer.
2. Donner une instruction contenant un appel à votre fonction (par exemple pour la tester).

Exercice 2. Un nombre premier est un entier qui n'a pas d'autre diviseur que 1 et lui-même.

1. Réaliser une fonction **estDiviseur** qui prend en argument deux entiers et renvoie `true` si le premier entier est un multiple du deuxième, `false` sinon. On rappelle que `a % b` est le reste de la division de `a` par `b`.
2. Réaliser une fonction **estPremier** qui prend en argument un entier et renvoie `true` si cet entier est un nombre premier, `false` sinon. Vous devez utiliser la fonction **estDiviseur**.
3. Réaliser une fonction **premiersBornes** qui prend en argument un entier `n` et qui affiche tous les nombres premiers inférieurs à `n`. Vous devez utiliser la fonction **estPremier**.
4. Réaliser une fonction **premiersPremiers** qui prend en argument un entier `n` et qui affiche les `n` premiers nombres premiers. Vous devez utiliser la fonction **estPremier**.
5. Écrire un programme qui demande à l'utilisateur un entier `n` et qui affiche les `n` premiers nombres premiers en faisant appel à la fonction **premiersPremiers**.

Exercice 3.

Écrire une fonction **lectureVB** (pour lectureValeurBornée) avec deux paramètres, un `min` et un `max`, qui demande à l'utilisateur d'entrer la valeur d'une variable réelle comprise entre ces deux bornes, puis qui renvoie cette valeur. La fonction doit afficher un message d'erreur si la valeur saisie n'est pas comprise entre les bornes et re-demander à l'utilisateur une nouvelle saisie jusqu'à pouvoir renvoyer une valeur correcte. Par exemple on peut utiliser cette fonction pour demander de saisir un nombre d'heures en imposant qu'il soit compris entre 0 et 23.

Exercice 4.

Une banque fait un prêt à une personne pour un montant total de s_0 euros. Cette personne rembourse chaque mois un montant fixe r et paye en plus un intérêt variable $i = i_m * s$ où i_m est le taux d'intérêt mensuel fixe et s la somme restant à rembourser (avant déduction du remboursement mensuel).

Écrire un programme qui demande à l'utilisateur le montant emprunté (s_0), le montant remboursé par mois (r) et le taux d'intérêt mensuel (i_m , qui doit être compris entre 0,1 % et 1,5 % pour ne pas dépasser le taux d'usure annuel fixé à 19,96 %), puis qui calcule et affiche la durée du remboursement et la somme des intérêts versés. Pour les lectures, vous pourrez utiliser la fonction **lectureVB** écrite précédemment quand cela vous paraîtra pertinent.

Exercice 5.

Réaliser un programme qui lit deux entiers positifs a et b tels que $a \leq b$ et un réel x , puis calcule $\sum_{i=a}^b \frac{x^i}{i!}$

1. en utilisant les fonctions **factorielle** et **puissance**, que vous réaliserez ;
2. en n'utilisant pas ces fonctions (et en réalisant un algorithme plus efficace en nombre d'opérations effectuées).

Exercice 6.

Soit le programme ci-dessous qui n'est pas un exemple de bonne programmation. Précisez la portée de chacune des variables et donnez les valeurs affichées.

```
(1) #include <iostream>
(2) using namespace std;
(3) int x = 10;
(4) int main() {
(5)     { int x = 5;
(6)         cout << "Un, x vaut :" << x << endl;
(7)     }
(8)     { cout << "Deux, x vaut :" << x << endl;
(9)     }
(10)    for (int x = 0; x < 3; x++) {
(11)        cout << "Trois, x vaut :" << x << endl;
(12)    }
(13)    cout << "Quatre, x vaut :" << x << endl;
(14)    for (x = 0; x < 3; x++) {
(15)        cout << "Cinq, x vaut :" << x << endl;
(16)    }
(17)    cout << "Six, x vaut :" << x << endl;
(18)    return 0;
(18) }
```

Exercice 7.

Donnez un programme qui pour tout entier strictement positif affiche en sortie son écriture en chiffres romains. Vous pourrez dans un premier temps considérer le système d'écriture simple qui, par exemple, à 1997 associe MDCCCCLXXXVII ; puis considérer le système qui, toujours par exemple, à 1997 associe MCMXCVII.