

CI 2: Boucles et tableau

l.cuvillon@unistra.fr

Télécom Physique Strasbourg — September 27, 2021

Consignes

Créer un fichier source par exercice.

Exercice 4

Soit le code suivant pour calculer le prix de vente brut (avec la TVA que paye le commerçant) pour une entreprise :

```
int main()
{
    int TVA=20;      /*en 0< pour cent entier <100*/
    double prix_brut, prix_net;
    ...
    prix_brut = prix_net + prix_net * (TVA/100);
    ...
}
```

1. Compléter le code précédent pour -demander à l'utilisateur de saisir au clavier un prix_net et -afficher dans le terminal le prix_net et prix_brut. Exécuter.
2. Le résultat obtenu n'est pas correct. Modifier l'instruction (TVA/100), pour obtenir un résultat correct sans modifier les types des variables.

Note: 2 solutions possibles, dont l'utilisation de l'opérateur de cast.

1 Les tests en langage C

Les instructions **if ... else** et **switch** (diapositives 128 à 139 du cours) permettent d'exécuter des instructions différentes en fonction du test d'une condition.

Exercice 5

1. Écrire un programme qui demande la saisie d'une variable de type entier et indique à l'utilisateur si celui-ci est pair ou impair. Pour le test, on peut utiliser l'opérateur modulo/reste de la division entière.
2. Ajouter un test qui indique à l'utilisateur par un message si celui-ci est strictement positif, strictement négatif ou nul.

Exercice 6 (Voyelles, consonnes)

Suite à la saisie par l'utilisateur d'une lettre minuscule non accentuée, indiquez à l'utilisateur si celle-ci est une voyelle ou une consonne.

Pour n'utiliser qu'un seul test `if ... else`, trouver une condition adéquate utilisant les relations logiques ET (`&&`), OU (`||`) :

```
if (      )
{ printf("voyelle\n"); }
else
{ ... }
```

2 Les boucles en langage C

Les instructions `while` et `for` (diapositives 142 à 156) permettent de répéter l'exécution d'un bloc d'instructions tant qu'une condition est valide. Avec les tests, elles constituent les instructions de contrôle du flux d'instructions.

Exercice 7 (Table de multiplication)

Écrire un programme qui affiche toute la table de multiplication (du 1er au 20e multiple) pour une valeur saisie au clavier. Utiliser une boucle `while` contenant entre autres une instruction `printf()`.

Exemple d'affichage pour la saisie 7:

```
1 * 7 = 7
2 * 7 = 14
...
20 * 7 = 140
```

Exercice 8 (Suite arithmétique)

Écrire un programme qui pour un entier n saisi au clavier, calcul (cf. calcul de la factorielle dans le cours) itérativement par une boucle `for()` la somme S :

$$S = 1 + 2 + 3 + \dots + n = \sum_{i=0}^n i$$

i

Note: Cette méthode de calcul itératif a une complexité algorithmiquement $O(n)$. Le temps d'exécution est linéairement proportionnel à la valeur n . Il existe une méthode de calcul en $O(1)$, à temps d'exécution constant, bien connue : $S = (n * (n + 1)) / 2$.