

Exercices en salle 1 *Bases de la programmation, partie 1*

1 Exercices rapides d'appropriation

1. À quoi sert le compilateur et que fait-il ?
2. Quel est le nom de la fonction principale d'un programme ? Par quoi se finit-elle toujours ?
3. Déclarez une variable de type `int`. Ensuite, donnez-lui la valeur 20.
4. Déclarez une variable de type `int`, nommée `value` et initialisée à `-5`.
5. Vous disposez déjà de la variable `int number = 42;`. Écrivez la ligne de code permettant d'afficher la valeur de la variable.
6. Quelle est la valeur de l'opération `13 / 4` ?
7. Quelle est la valeur de l'opération `13 % 4` ?
8. Écrivez la condition qui teste si la valeur de la variable `v` est strictement plus petite que 7.
9. Écrivez un programme qui affiche le texte *Bravo* à l'écran si la variable `A` est égale à 5, et *Dommage* si ce n'est pas le cas.
10. Écrivez une boucle `for` qui fait aller la variable `i` de 0 à 3 (inclus).
11. Écrivez la même boucle mais sous forme de `while`.
12. Écrivez une boucle `for` qui fait aller la variable `k` de 0 à 10 (inclus), mais par pas de deux.

2 Instruction `if-else`

Voici un extrait de code.

```

1  int value = ...;
2
3  if (...)
4  {
5      printf ("La valeur de value est un nombre pair");
6  }
7  else
8  {
9      printf ("La valeur de value est un nombre impair");
10 }
```

1. Complétez la condition du `if` pour tester si la valeur de `value` est paire.
2. Comment adapter votre code pour que le `if` teste si la valeur de `value` est impaire.
3. On peut remarquer qu'il y a une partie de code qui est répétée, laquelle ? Comment pouvez-vous simplifier le code pour éviter cette répétition ?

3 Boucles for et while

1. Voici une boucle qui effectue des opérations sur une variable **a**.

```
1 int i;
2 int a = 0;
3
4 for (i = 5; i > 0; i = i - 1)
5 {
6     a = a + i;
7 }
8
9 printf ("a vaut %d et i vaut %d", a, i);
```

Exécutez cette boucle manuellement en construisant le tableau des valeurs des différentes variables à chaque itération. Quelles seront les valeurs de **a** et **i** qui seront affichées à la fin ?

2. Si le code précédent avait été modifié comme suit, quelle sera la différence ?

```
1 int i;
2 int a = 0;
3
4 for (i = 5; i > 0; i = i + 1)
5 {
6     a = a + i;
7 }
8
9 printf ("a vaut %d et i vaut %d", a, i);
```

3. Voici maintenant une boucle **while**.

```
1 int i;
2
3 while (i % 2)
4 {
5     i = i - 1;
6 }
7
8 printf ("Terminé !");
```

- (a) Pour quelles valeurs assignées à **i** la boucle ne sera-t-elle pas exécutée ?
 - (b) Sur base de votre réponse à la question précédente, pouvez-vous prédire combien de fois, au maximum, la boucle sera exécutée ?
4. Écrivez une boucle qui permet d'afficher toutes les puissances de 10, en partant de 1000000 jusqu'à 1.