

Exercice 1

Pour chacun des deux programmes suivants, le recopier en l'indentant, le tracer, et dire ce qui s'affiche à l'exécution. Expliquer la différence essentielle entre les deux programmes, et dire lequel des deux correspond le mieux aux spécifications annoncées dans les commentaires.

```
/* Programme 1 */
/* Affichage des carre et des cube des petits entiers */
#include<stdio.h>

main(){
  int a, carre, cube;
  printf("Nombre\t Carre\t cube\n");
  //Point d'observation 1
  for(a=1; a<=5; a++){
    carre = a*a;
    cube = carre*a;
    printf("%i\t", a);
    printf("%i\t", carre);
    printf("%i\n", cube);
    //Point d'observation 2
  }
}

/* Programme 2 */
/* Affichage des carre et des cube des petits entiers */
#include<stdio.h>

main(){
  int a, carre, cube;
  printf("Nombre\t Carre\t cube\n");
  //Point d'observation 1
  for(a=1; a<=5; a++)
    carre = a*a;
  cube = carre*a;
  printf("%i\t", a);
  printf("%i\t", carre);
  printf("%i\n", cube);
  //Point d'observation 2
}
```

Exercice 2

Écrire un programme en langage C qui demande à l'utilisateur un nombre entier a et qui en retour affiche la table de multiplication pour les nombres entre 1 et a . Par exemple, si l'utilisateur

choisit 5, le programme affichera :

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

Exercice 3

Écrire un programme qui demande un caractère à l'utilisateur, et qui affiche en retour "le caractère est un chiffre", "le caractère est une lettre minuscule", ou "le caractère est une lettre majuscule", selon la valeur du caractère. Si le caractère n'appartient à aucune des catégories ci-dessus, le programme doit afficher le message "autre sorte de caractère".

Exercice 4

QCM. NE PAS RENDRE LE SUJET MAIS RÉPONDRE SUR FEUILLE : 1a, 2d, 3b ... par exemple.

Attention : Pour chaque question, une et une seule réponse est correcte. Une bonne réponse rapporte des points, une mauvaise réponse coûte des points, l'absence de réponse est neutre.

1. Dans la ligne de programme `printf("Le nombre vaut %i.", a);`
 - a. `%` est un opérateur permettant de calculer un reste
 - b. On utilise `%i` car `a` est de type `int`
 - c. Il y a une faute de syntaxe
 - d. Il y a une faute d'orthographe
2. L'expression `a<=1`
 - a. Réalise une affectation
 - b. Diminue de 1 la valeur de `a`
 - c. A pour valeur VRAI si $a \leq 1$ et FAUX sinon
 - d. Utilise les opérateurs `<` et `=`
3. `if (a<5) printf("Bonjour");`
`a=a+1;`
 - a. Affiche bonjour et augmente la valeur de `a` quelque soit `a`
 - b. Augmente la valeur de `a` quelque soit `a`
 - c. Affiche bonjour quelque soit `a`
 - d. N'affiche pas bonjour et n'augmente pas la valeur de `a` quelque soit `a`
4. Après `char c; c='a'; c=c+1;`
 - a. `c` vaut `'A'`
 - b. `c` vaut `'a'`
 - c. `c` vaut `'b'`
 - d. Un message d'erreur s'affiche

5. `printf("%c", 'A')`
- Affiche le caractère 'A'
 - Affiche le code ASCII du caractère 'A'
 - Affiche le code ASCII du caractère stocké dans la variable *A*
 - Affiche le caractère dont le code ASCII est stocké dans la variable *A*
6. `printf("%i", 'A')`
- Affiche le caractère 'A'
 - Affiche le code ASCII du caractère 'A'
 - Affiche le code ASCII du caractère stocké dans la variable *A*
 - Affiche le caractère dont le code ASCII est stocké dans la variable *A*
7. `printf("%c", A)`
- Affiche le caractère 'A'
 - Affiche le code ASCII du caractère 'A'
 - Affiche le code ASCII du caractère stocké dans la variable *A*
 - Affiche le caractère dont le code ASCII est stocké dans la variable *A*
8. `printf("%i", A)`
- Affiche le caractère 'A'
 - Affiche le code ASCII du caractère 'A'
 - Affiche le code ASCII du caractère stocké dans la variable *A*
 - Affiche le caractère dont le code ASCII est stocké dans la variable *A*
9. `if (a%2) printf("bonjour");`
- Affiche bonjour quand *a* est un entier pair
 - Affiche bonjour quand *a* est un entier impair
 - Déclenche le message d'erreur `invalid lvalue in assignment`
 - N'affiche rien (quelque soit le type et la valeur de *a*)
10. `if (a%2 == 0) printf("bonjour");`
- Affiche bonjour quand *a* est un entier pair
 - Affiche bonjour quand *a* est un entier impair
 - Déclenche le message d'erreur `invalid lvalue in assignment`
 - N'affiche rien (quelque soit la valeur de *a*)
11. `if (a%2 = 0) printf("bonjour");`
- Affiche bonjour quand *a* est un entier pair
 - Affiche bonjour quand *a* est un entier impair
 - Déclenche le message d'erreur `invalid lvalue in assignment`
 - N'affiche rien (quelque soit la valeur de *a*)
12. En langage C, `++` est
- Un opérateur de décrément
 - Un opérateur d'accrétion
 - Un opérateur d'incrément
 - Un opérateur de décrétinisation