

Nom		Prenom	
Class/ID		Date	

QCM #2 - Language C

- Quel est le rôle principal du fichier `run.sh` ?
 - Lancer Vim directement
 - Créer un environnement virtuel Python
 - Démarrer un conteneur Docker configuré pour le C
 - Compiler automatiquement tous les fichiers C
- Quelle commande clone correctement ce dépôt GitHub ?
 - `git init https://github.com/Tekina-Ecole-IT-Bxl/01GCC-2025`
 - `git start https://github.com/Tekina-Ecole-IT-Bxl/01GCC-2025`
 - `git clone https://github.com/Tekina-Ecole-IT-Bxl/01GCC-2025.git`
 - `git load 01GCC-2025`
- Quel est le mot de passe par défaut pour se connecter à l'environnement Docker
 - `dev`
 - `gcc`
 - `root`
 - Aucun, c'est une connexion sans mot de passe
- Quelle commande donne les droits d'exécution à `run.sh` ?
 - `chmod 775 run.sh`
 - `execute run.sh`
 - `make run.sh`
 - `chmod +x run.sh`

5. Que fait cette condition ?

...

if (x = 5)

...

- a. Compare si x vaut 5
- b. Affecte 5 à x, puis entre dans la condition
- c. Provoque une erreur
- d. Compare 5 à une autre variable implicite

6. Que contient principalement le dossier 1-Basics ?

- a. Des scripts Bash pour gérer Docker
- b. Des exercices simples pour manipuler les bases du C
- c. Un jeu Snake codé en assembleur
- d. Un clone de printf

7. Quelle commande permet de compiler un fichier main.c ?

- a. `compile main.c -o main`
- b. `cc main.c`
- c. `gcc main.c -o main`
- d. `launch main`

8. Quel est le rôle d'un pointeur en C ?

- a. Créer une nouvelle variable globale
- b. Remplacer une fonction
- c. Stocker l'adresse d'une variable
- d. Allouer de la mémoire automatiquement

9. Que fait ce code ?

```
```c
```

```
 int *p;
```

```
 int x = 10;
```

```
 p = &x;
```

```
 printf("%d", *p);
```

```
```
```

- a. Affiche 0
- b. Affiche une adresse mémoire
- c. Affiche 10
- d. Erreur de compilation

10. Quel est le rôle du mot-clé return dans une fonction C ?

- a. Arrête le programme
- b. Renvoie une valeur au programme appelant
- c. Relance la fonction
- d. Crée une variable temporaire

11. Quelle boucle est infinie en C ?

- a. for (i = 0; i < 1; i++)
- b. while (1)
- c. do { break; } while(1);
- d. for (int i = 0; i < 10; i--)

12. Que fait le code suivant ?

```
```c
```

```
int func(int x) {
 if (x <= 0) return 0;
 return x + func(x - 1);
}
```
```

- a. Affiche 0
- b. Calcule la factorielle
- c. Calcule la somme de 1 à x
- d. Crée une boucle infinie

13. Quelle est la différence entre == et = ?

- a. = compare, == affecte
- b. == compare, = affecte
- c. Les deux sont identiques
- d. == est utilisé uniquement avec des char

14. Pourquoi ce code est-il dangereux ?

```
```c
```

```
char *str = "abc";
str[0] = 'A';
```
```

- a. On modifie une constante
- b. Le code est parfait
- c. Le pointeur est null
- d. C est un langage typé dynamique

15. Que fait ce code ?

```
```c
```

```
int tab[3] = {1, 2, 3};
printf("%d", *(tab + 1));
```
```

- a. Affiche 1
- b. Affiche 2
- c. Affiche l'adresse de tab
- d. Erreur de compilation

Bonus:

1. Quel est le résultat cet fonction?

```
```c
```

```
#include <stdio.h>
```

```
void mystere(int *a, int *b) {
 if (*a > *b)
 *a = *b;
 else
 *b = *a;
}
```

```
int main() {
 int x = 4, y = 2;
 mystere(&x, &y);
 printf("%d %d\n", x, y);
 return 0;
}
```

```
```
```

A. 2 2

B. 4 4

C. 4 2

D. 2 4

2. Quel est l'effet du programme ?

```
```\n#include <stdio.h>\n\nint* mystere() {\n    int x = 42;\n    return &x;\n}\n\nint main() {\n    int *p = mystere();\n    printf("%d\\n", *p);\n    return 0;\n}
```

- A. Il affiche 42
- B. Il affiche une valeur aléatoire ou provoque un crash
- C. Il affiche 0
- D. Il affiche l'adresse mémoire de x

3. Écrire un programme en C qui affiche les nombres de 1 à 100, avec une règle conditionnelle :

- Si le nombre est un **multiple de 3**, afficher "Fizz" à la place du nombre.
- Si le nombre est un **multiple de 5**, afficher "Buzz" à la place du nombre.
- Si le nombre est un **multiple à la fois de 3 et 5**, afficher "FizzBuzz".
- Sinon, afficher simplement le **nombre lui-même**.

