Identification /5

Prénom: .....

Nom: .....

Code permanent: .....

Veuillez choisir **une seule réponse** par question, avec une croix (×) ou un crochet (✓).

Q1. Qu'est-ce qui sera affiché par le code suivant?

```
int arrondir(double x) {
  if (x < 0) return x - 0.5;
  return x + 0.5;
}

int main(void) {
  printf("%d ", arrondir(2.7));
  printf("%d ", arrondir(-2.7));
  printf("%d\n", arrondir(7 / 2));
  return 0;
}
```

- ☐ `3 -3 4`
- ☐ `3 -4 4`
- ☐ `3 -3 3`
- ☐ `3 -4 3`

Q2. Lequel des prototypes suivants est valide?

- ☐ `int maximum(int a, b);`
- ☐ `int maximum(int a; int b);`
- ☐ `int maximum(int, int);`
- ☐ `maximum(int a, int b);`

Q3. Selon le format exigé dans le cours, que doit contenir la *docstring* de la fonction suivante, en plus de la phrase résumant ce qu'elle fait?

```
void afficher_ligne(char c, int n);
```

- ☐ Un seul `@param` décrivant `c` et `n`
- ☐ Un `@param` pour `c` et un pour `n`
- ☐ Un `@param` pour `c`, un pour `n` et un `@return`
- ☐ Un `@return` seulement

Q4. Qu'est-ce qui sera affiché par le code suivant?

```
int soustraire(int n, int k) {
  n = n - k;
  return n;
}

int main(void) {
  int n = 3, k = 4;
  int r = soustraire(k, n);
  printf("%d %d %d\n", n, k, r);
  return 0;
}
```

- ☐ `3 4 -1`
- ☐ `3 1 1`
- ☐ `-1 4 -1`
- ☐ `3 4 1`

Q5. Écrivez la définition de la fonction

```
int nombre_diviseurs(int n);
```

précédée de sa *docstring*, dans le format exigé dans le cours. La fonction reçoit un entier `n` supérieur ou égal à 1 et *retourne* le nombre de diviseurs positifs de `n`. Par exemple, `nombre_diviseurs(12)` retourne 6 (les diviseurs sont 1, 2, 3, 4, 6 et 12) et `nombre_diviseurs(7)` retourne 2. La fonction ne doit rien afficher.

Note: utilisez le verso.