Identification /5

Prénom: .....

Nom: .....

Code permanent: .....

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

Considérez le programme `prog.c` suivant:

```
#include <stdio.h>

void f(int* a, int b) {
  *a = ++b;
  b = (*a)--;
}

int main(void) {
  int a = 1, b = 2;
  f(&a, b);
  printf("%d\n%d\n", a, b);
  return 0;
}
```

Q1. Qu'est-ce qui sera affiché sur la première ligne de la sortie standard lorsqu'on exécute le programme `prog.c`?

☐ 0 ☐ 1 ■ 2 ☐ 3

Q2. Qu'est-ce qui sera affiché sur la deuxième ligne de la sortie standard lorsqu'on exécute le programme `prog.c`?

☐ 0 ☐ 1 ■ 2 ☐ 3

Considérez le programme `str.c` suivant:

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

int main(void) {
  char s[] = "quiz", t[5];
  t[0] = 0; strcat(t, s);
  printf("%lu\n%lu\n", sizeof(t),
                       strlen(t));
  return 0;
}
```

Q3. Qu'est-ce qui sera affiché sur la première ligne de la sortie standard lorsqu'on exécute le programme `str.c`?

☐ 0 ☐ 1 ☐ 4 ■ 5

Q4. Qu'est-ce qui sera affiché sur la deuxième ligne de la sortie standard lorsqu'on exécute le programme `str.c`?

☐ 0 ☐ 1 ■ 4 ☐ 5

Q5. Qu'est-ce qui sera affiché sur la sortie standard si on exécute le programme suivant?

```
#include <stdio.h>

int main(void) {
  int c = 0;
  for (int i = 0; i < 5; ++i) {
    for (int j = 4; j >= 0; --j) {
      if ((i + j) % 3 == 0)
        break;
      ++c;
    }
  }
  printf("%d\n", c);
  return 0;
}
```

☐ 0 ■ 6 ☐ 17 ☐ 25

Voir explications page suivante

Q1. Le premier argument de la fonction `f` est une adresse, ce qui correspond à un pointeur lorsqu'on utilise le paramètre `int* a`. L'instruction `*a = ++b` incrémente d'abord la valeur (locale) de `b` à `3`, de sorte que la valeur `*a` passe à `3`. Ensuite, l'affectation `b = (*a)--` modifie la valeur locale de `b`, puis décrémente `*a` en la ramenant à `2`. Comme on a passé une adresse, la modification persiste, c'est-à-dire que la variable `a` de la fonction `main` est bien égale à `2` après avoir appelé `f`.
Q2. Le deuxième argument de la fonction `f` est passé par valeur. Cela signifie donc que le paramètre `int b` est modifié localement lorsqu'on exécute les instructions de la fonction `f`. La valeur `int a` de la fonction `main` reste donc inchangée par l'appel.
Q3. L'opérateur `sizeof` appliqué à un tableau retourne le nombre d'octets occupé par ce tableau. Comme `t` est un tableau de 5 caractères, sa taille est égale à `5 * sizeof(char)`, qui vaut bien 5, puisque `sizeof(char)` vaut 1.
Q4. L'affectation `t[0] = 0` fait en sorte que `t` devient une chaîne de caractères vide bien formée (rappelons que `'\0'` est le caractère nul, qui vaut 0 numériquement, c'est-à-dire que `t[0] = 0` est équivalent à `t[0] = '\0'`). Ensuite, l'appel de la fonction `strcat` concatène `"quiz"` à la suite de la chaîne vide, ce qui fait que `t` contient la chaîne `"quiz"`, qui est bien formée. La fonction `strlen` retourne la longueur de cette chaîne de caractère, qui est bien 4. Ne pas oublier que le caractère nul se trouvant à la fin de la chaîne n'est pas comptabilisé.
Q5. Afin de ne pas se tromper, la méthode la plus prudente pour répondre à cette question consiste à faire une trace détaillée en la présentant sous forme de tableau:

`c`	0	1		2	3			4		5	6
`i`	0		1			2	3		4		
`j`	4	3	4	3	2	4	4	3	4	3	2
`i + j`	4	3	5	4	3	6	7	6	8	7	6
`(i + j) % 3`	1	0	2	1	0	0	1	0	2	1	0

Plus spécifiquement, chaque ligne correspond à une expression apparaissant dans le programme et à l'évolution des valeurs de ces expressions au fur et à mesure. On laisse l'entrée du tableau vide si la valeur est restée inchangée.