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 sur la sortie standard par le fragment de code suivant?

```
int t[] = {5, 10, 15, 20, 25, 30};
int *p = &t[4], *q = t;
q += 2;
printf("%d %d %d %d\n", *q, p[1],
       (int)(p - q), *(p - 1) + 1);
```

☐ `15 30 8 21`
☐ `15 30 2 21`
☐ `15 26 2 21`
☐ `15 30 2 25`

Q2. Considérez le programme `joindre.c` suivant:

```
int main(int argc, char *argv[]) {
  if (argc < 3) {
    fprintf(stderr,
            "usage: joindre A B\n");
    return 1;
  }
  printf("%s-%s\n", argv[2], argv[1]);
  return 0;
}
```

Une fois le programme compilé, qu'est-ce qui sera affiché dans le terminal par les commandes suivantes?

```
$ ./joindre un deux > sortie.txt
$ echo $?
```

☐ `deux-un`, puis `0`
☐ `0` seulement
☐ `un-deux`, puis `0`
☐ `usage: joindre A B`, puis `1`

Q3. Considérez la fonction suivante:

```
int compter(void) {
  static int n = 0;
  int k = 0;
  ++n;
  ++k;
  return 10 * n + k;
}
```

Que retourneront trois appels successifs à `compter()`, dans l'ordre?

☐ `11 11 11`
☐ `10 20 30`
☐ `11 22 33`
☐ `11 21 31`

Q4. Qu'affichera le fragment de code suivant?

```
struct Etudiant {
  char nom[10]; int note;
};

int compare(const void *a,
            const void *b) {
  const struct Etudiant *x = a, *y = b;
  return y->note - x->note;
}
// ...
struct Etudiant t[] = {
  {"Alice", 72}, {"Bob", 85},
  {"Carl", 64}, {"Dina", 91}};
qsort(t, 4, sizeof(struct Etudiant),
      compare);
printf("%s %s\n", t[0].nom, t[3].nom);
```

☐ `Dina Carl`
☐ `Alice Dina`
☐ `Carl Dina`
☐ `Bob Carl`

Q5. Complétez le corps de la fonction suivante, qui retourne l'élément du tableau `t`, de taille `n >= 1`, pour lequel `cle` retourne la plus grande valeur (le premier, en cas d'égalité):

```
int maximum_selon(
  const int *t, unsigned int n,
  int (*cle)(int)) {
  // TODO
}
```

Exemple: si `absolue` retourne la valeur absolue de son argument, alors

```
int v[] = {3, -7, 5};
maximum_selon(v, 3, absolue); // -7
```

Note: utilisez le verso.