

Identification

Prénom:
Nom:
Code permanent:

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

Q1. Quelle extension est utilisée pour nommer les fichiers contenant une bibliothèque dynamique en Unix?

- | | |
|-------------------------------|-------------------------------|
| <input type="checkbox"/> .a | <input type="checkbox"/> .lib |
| <input type="checkbox"/> .dll | <input type="checkbox"/> .so |

Q2. Quelle option du compilateur GCC sert à spécifier des répertoires supplémentaires dans lesquels se trouvent des fichiers binaires de bibliothèques tierces?

- | | |
|-----------------------------|-----------------------------|
| <input type="checkbox"/> -I | <input type="checkbox"/> -v |
| <input type="checkbox"/> -L | <input type="checkbox"/> -W |

Q3. Quelle commande permet de lister les répertoires dans lesquels se trouvent les fichiers d'en-tête de la bibliothèque Cairo?

- `cairo --headers`
- `gcc -I cairo`
- `pkg-config --cflags cairo`
- `pkg-config --libs cairo`

Q4. Quelle expression de l'outil Make permet de lister les noms de tous les fichiers du répertoire courant qui commencent par `test` et qui ont l'extension `c`?

- `$(patsubst test%.c)`
- `$(patsubst test*.c)`
- `$(wildcard test%.c)`
- `$(wildcard test*.c)`

Q5. Parmi les comportements suivants, lequel correspond à un programme qui n'est pas robuste?

- Il ne retourne pas la valeur attendue
- Il ne retourne aucune valeur
- Il peut boucler à l'infini
- Il retourne la valeur 1

Q6. Quelle option du compilateur GCC permet de désactiver les assertions d'un programme C?

- `--no-debug`
- `-DNDEBUG`
- `--no-assert`
- `--release`

Q7. Parmi les éléments suivants, lequel caractérise la programmation défensive?

- Accepter les postconditions les plus faibles
- Exiger les préconditions les plus strictes
- Mettre en place des assertions
- Ne pas faire confiance à l'utilisateur

Q8. Rappelons l'interface d'une pile en C vue en classe:

```
#include <stdbool.h>

// Types
struct StackNode {
    char value;
    struct StackNode* next;
};

typedef struct {
    struct StackNode* first;
    unsigned int size;
} Stack;

// Fonctions
void stack_initialize(Stack *s);
bool stack_is_empty(const Stack *s);
void stack_push(Stack *s, char value);
char stack_pop(Stack *s);
void stack_delete(Stack *s);
```

Parmi les opérations suivante, laquelle a comme postcondition `!stack_is_empty(s)`?

- `stack_initialize`
- `stack_push`
- `stack_pop`
- `stack_delete`