

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?

- | | |
|-------------------------------|---|
| ☐ .a | ☐ .lib |
| ☐ .dll | ☒ .so |

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

- | | |
|--|-----------------------------|
| ☐ -I | ☐ -v |
| ☒ -L | ☐ -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 |

[Voir explications page suivante](#)

Q1. L'extension `.a` est utilisée pour les bibliothèques statiques en Unix. Les extensions `.lib` et `.dll` sont des extensions utilisées dans les systèmes Windows, pour les bibliothèques statiques et dynamiques, respectivement.

Q2. L'option `-I` sert à spécifier des chemins dans lesquels se trouvent des fichiers d'en-tête (et non des fichiers binaires). L'option `-w` est utilisée pour activer ou désactiver certains avertissements du compilateur, alors que l'option `-v` permet d'afficher des informations sur la version du compilateur.

Q3. La commande `cairo --headers` n'existe pas. La commande `gcc -I cairo` indique au compilateur de considérer le sous-répertoire `cairo` du répertoire courant dans sa recherche de fichiers d'en-tête (rien ne garantit qu'il contienne les fichiers d'en-tête de Cairo). Finalement, la commande `pkg-config` facilite la recherche des répertoires qui contiennent des fichiers d'en-tête et autres options de compilation (`--cflags`) d'une bibliothèque tierce, de même que les fichiers binaires et autres options d'édition de liens (`--libs`).

Q4. La fonction `wildcard` de Make permet de lister des chemins vers des fichiers à partir du répertoire courant, alors que la fonction `patsubst` sert à substituer un motif par un autre dans un chemin. Le caractère `*` est un *joker* qui a le même comportement que `*` pour le *glob*, alors que le caractère `%` sert plutôt à identifier le motif sur lequel appliquer une substitution.

Q5. Lorsqu'un programme tombe dans une boucle infinie non souhaitée, c'est sa robustesse qui est en cause. Le premier choix est plutôt relié à la notion de *justesse*. Le troisième choix entraînera un avertissement du compilateur, qui choisira une valeur arbitraire à retourner si on décide d'ignorer l'avertissement (ce n'est pas un problème de robustesse, mais plutôt de compilation). Finalement, le dernier choix n'est pas problématique en soi: un programme peut retourner 1 pour signaler une erreur d'utilisation (ce n'est donc pas un problème de robustesse ou de justesse).

Q6. L'option `-DNDEBUG` permet de définir le symbole `NDEBUG`, ce qui a pour effet de désactiver les assertions. Les trois autres options n'existent pas.

Q7. Les trois autres éléments caractérisent plutôt la programmation par contrats.

Q8. Attention, on parle ici de *postcondition*, c'est-à-dire une propriété qui est vérifiée à la fin de l'opération, par opposition à une *précondition*, qui est vérifiée avant d'appliquer l'opération. Or, la seule opération qui garantisse que la pile soit non vide après avoir été appliquée est `stack_push`.