

Exercices: Fonctions

Introduction au langage C – Coda 1 ère année – Septembre 2025

- 1 – Écrire une fonction qui calcule la factorielle d'un nombre et retourne le résultat
- 2 - Écrire un programme qui teste si un nombre est premier. La fonction doit retourner 0 si c'est le cas et 1 sinon
- 3 – Écrire une fonction qui vérifie si une chaîne de caractère est un palindrome. La fonction doit retourner 0 si c'est le cas et 1 sinon.

Exercices graphiques

Les exercices suivants vont vous demander de coder des algorithmes qui affichent des formes de couleurs dans une interface graphique. Pour ce faire, nous vous fournissons des fonctions permettant de vous simplifier la programmation graphique. **Vous devrez utiliser le projet qui se trouve dans le zip du TP**

Installation des librairies nécessaires

Ces fonctions utilisent la librairie CSFML pour la gestion de l'interface graphique. Vous devez d'abord installer csfml dans votre environnement pour pouvoir faire l'exercice. Exécutez les 3 commandes suivantes qui vont installer SFML puis CSFML

```
sudo apt update  
sudo apt install libsFML-dev
```

```
sudo apt install libcsfml-dev
```

Description des fonctions disponibles

Pour les prochains exercices, vous devez utiliser le code qui se trouve dans le dossier **src**.

Vous y trouverez un fichier **main.c** qui affiche un carré vert

```
#include "library.h"  
#include <stdlib.h>  
#include <time.h>  
  
int main() {  
    create_square(0,255,0);  
  
    draw();  
    return 0;  
}
```

Ce code utilise une librairie externe `library.a` qui a la capacité de créer des fenêtres dans votre environnement UNIX et dessiner des carrés de couleur dessus. Cette librairie vous donne accès aux fonctions suivantes

```
void create_square(int red, int green, int blue);
void new_line();
void create_empty_square();
```

Les fonctions permettent les choses suivantes :

create_square: affiche un carré de couleur de 30 pixels avec les couleurs RGB définies en paramètres. Le premier appel à **create_square** dessine un carré en haut à gauche de la fenêtre. Si vous appelez une nouvelle fois la fonction, le prochain carré est affiché à côté sur la même ligne.

new_line: Permet de passer à la ligne. Les prochains carrés dessinés par **create_square** se trouveront sur la ligne suivante.

create_empty_square: Dessine un carré invisible et décale les prochains carrés affichés. Cette fonction permet essentiellement de faire des espaces avec des carrés vides.

Vous devrez placer votre code entre le début de la fonction `main` et l'appel à la fonction `draw`. Ne supprimez pas l'appel à la fonction `draw` car elle est responsable de la création de la fenêtre

```
int main() {
    // votre code

    draw();
    return 0;
}
```

Pour compiler votre programme C, au lieu de la commande `gcc` habituelle, il vous suffira d'utiliser la commande `make` à la racine du dossier `src`. Cela compilera automatiquement votre programme et créera un exécutable nommé `.main` que vous pouvez lancer comme d'habitude

```
TP5/src$ make
gcc -Wall -I. main.c -L. -lsquares -lcsfml-graphics -lcsfml-window -o main
TP5/src$ ./main
```

Note: Nous expliquerons plus tard dans le module à quoi servent tout les fichiers qui se trouvent dans le dossier ainsi que la commande make.

Exercices

Pour les exercices suivants, vous ne devez modifier que le code qui se trouve dans le fichier main.c et n'utiliser que les 3 fonctions décrites plus haut.

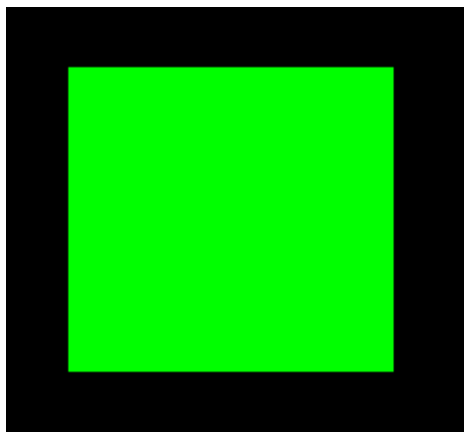
Il ne vous est pas demandé de comprendre comment fonctionne la librairie CSFML ou la librairie externe librairie.c et .h. Pour votre curiosité, le code est fourni et vous pouvez le lire.

Il est important, en tant que futurs développeurs, de savoir utiliser un code écrit par un tiers. Il est également essentiel de faire preuve d'abstraction et de vous projeter dans un projet dont vous ne connaissez pas encore le contexte.

4 - Dessiner une ligne de 5 blocs bleus



5 - Dessiner un carré vert de taille 5x5

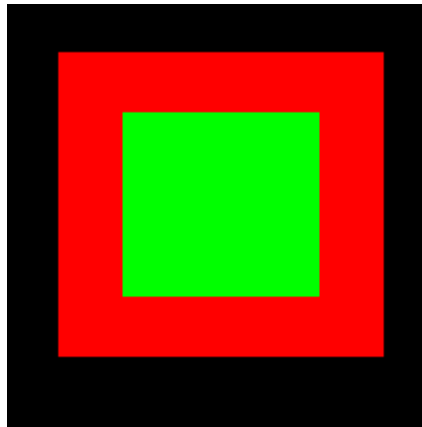


6 - Écrire une fonction create_line qui prend 4 paramètres :

Le nombre bloc sur la ligne
les 3 composantes de couleurs

La fonction devra dessiner automatiquement une ligne en se basant sur les paramètres

7 – Reproduire la forme suivante : Réutiliser les fonctions précédentes si possible



8 – Le code suivant permet de générer un chiffre aléatoire entre 0 et 255

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

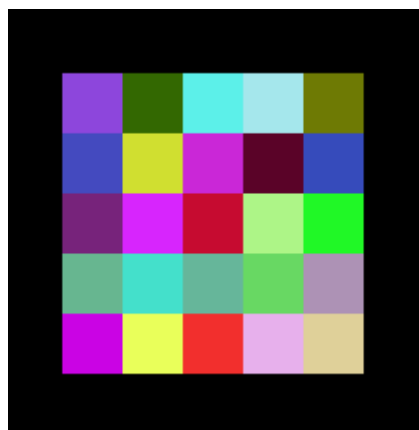
int main() {
    srand(time(NULL));

    // generate a random number between 0 and 255
    int random_number = rand() % 256;
    printf("Chiffre aléatoire : %d\n", random_number);

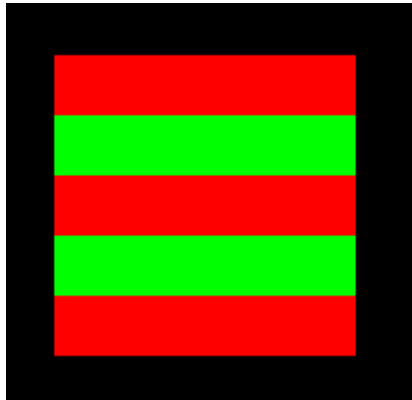
    return 0;
}
```

note: srand(time(NULL)) ne doit être appelé qu'une fois dans le programme

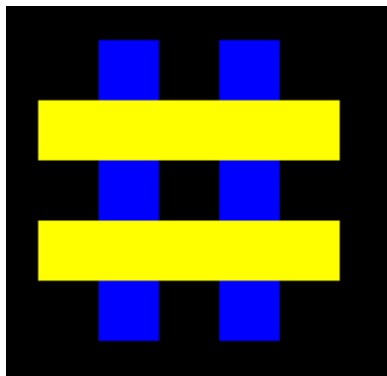
Dessinez un carré 5x5 avec des couleurs aléatoires



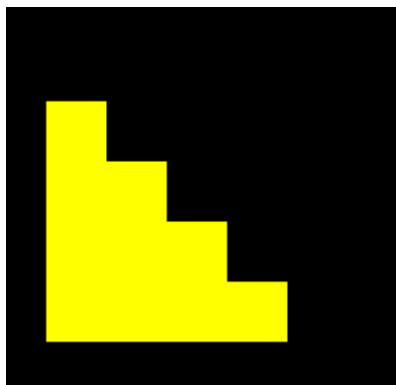
9 – Reproduire la forme suivante: Vous devez vous servir de l'opérateur %



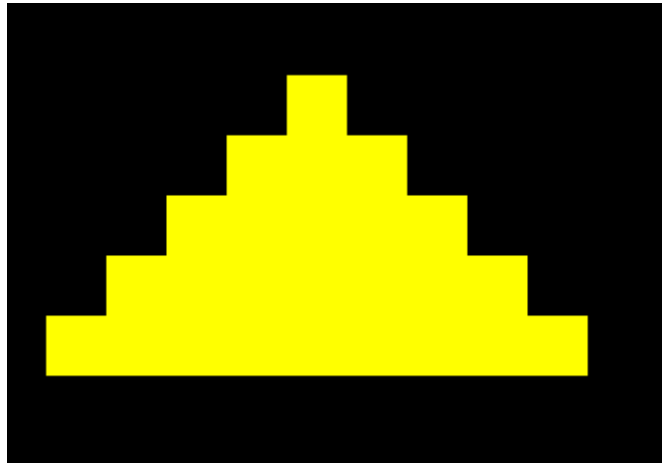
10 – Reproduire la forme suivante:



11 – Reproduire la forme suivante:



12 – Reproduire la forme suivante:



13 – Reproduire la forme suivante:

