

LIFAP1 – TP3 : Passage de paramètres / GRAPIC

Objectifs : Mode de passage des paramètres : données, données/résultats
Prise en main de GRAPIC

1. Kezako (Compréhension : fonction/procédure, paramètres et appels)

a. Que fait le programme ci-dessous ? Réfléchissez et écrivez le texte que vous pensez voir apparaître à l'exécution de ce programme :

```
#include <iostream>
using namespace std;

void proc_mult(int a, int b, int& ab)
{
    cout << "execution de la procedure proc_mult" << endl;
    ab = a*b;
}

int fonc_mult(int a, int b)
{
    cout << "execution de la fonction fonc_mult" << endl;
    return a*b;
}

void kezako(int x, int y, int r1, int& r2)
{
    proc_mult( x, y, r1);
    r2 = fonc_mult(x,y);
    cout << "A la fin de kezako r1=" << r1 << " r2=" << r2 <<endl;
}

int main(void)
{
    int a, y, res1, res2;
    a = 5;      y = 4;      res1 = 0;      res2 = 1;
    cout <<"Dans main avant kezako res1=" << res1 << " res2=" <<res2 <<endl;
    kezako( a, y, res1, res2);
    cout << "Dans main apres kezako res1=" << res1 << " res2="<<res2<< endl;
    return 0;
}
```

b. Téléchargez le programme Kezako à partir de la page du cours afin de vérifier si votre intuition est correcte. Sinon demandez une explication à votre encadrant de TP.

2. Passage de paramètres. Ecrire pour chaque exercice le sous-programme demandé ainsi que le programme principal permettant de le tester.

- Permutation circulaire de 3 variables.
- Nombre de combinaisons (fonction **ET** procédure).

3. Prise en main de l'outil GRaPiC : Graphics for Algo/Prog in C/C++

Pour réaliser les exercices demandés, reprenez vos notes du TD 5 et n'hésitez pas à consulter les tutoriels disponibles en ligne :

<http://liris.cnrs.fr/alexandre.meyer/grapic/html/index.html> .

a. Téléchargez la dernière version de GRaPiC et installez-la sur votre compte. Ouvrez ensuite le fichier Grapic/build/windows/grapic_workspace avec Codeblocks. Sélectionnez dans la partie gauche de la fenêtre le projet que vous souhaitez utiliser (LIFAP1_TP3) puis à l'aide du bouton droit, choisissez **Activate project**.

Vous trouverez une copie des résultats qu'on souhaite obtenir dans les questions suivantes à la fin de l'énoncé.

- b. Ecrivez un sous-programme qui affiche au centre de la fenêtre un carré dont la taille sera passée en paramètre. Vous utiliserez la procédure `GRaPiC Rectangle`.
- c. Réutilisez la procédure précédente pour afficher n (passé en paramètres) carrés imbriqués.
- d. Ecrivez un sous-programme qui dessine un cercle plein dont le rayon et la couleur sont passés en paramètres. Vous utiliserez pour cela `CircleFill`.
- e. En utilisant la procédure précédente, dessinez à présent une cible (alternance de cercles concentriques noirs et blancs).
- f. Dessinez une grille de carrés produisant un dégradé de couleur du vert (dans le coin supérieur gauche) au rouge (dans le coin inférieur droit).
- g. Ecrivez un sous-programme qui affiche une image dont la taille diminue au cours du temps.
- h. En utilisant 3 images différentes, générez une grille d'images positionnées aléatoirement.
- i. Sans utiliser la procédure `Circle` de `GRaPiC`, dessinez un cercle plein dont le rayon r est passé en paramètre et contenant le caractère c également passé en paramètre. En supposant que le centre du cercle est au centre de la fenêtre, vous utiliserez le calcul de la distance au centre de tout point de la fenêtre pour déterminer s'il appartient ou non au cercle.

Rappel : Soient $A (x, y)$ et $O (x_0, y_0)$ alors la distance entre A et O est $\sqrt{(x - x_0)^2 + (y - y_0)^2}$

