

LIFAP1 – TP5 : Passage de paramètres

Objectifs : Mode de passage des paramètres : données, données/résultats

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. Écrivez une procédure qui affiche une frise.

```
procedure afficherFrise(n,l,h : donnée Entier)
// n est le nombre de fois que se repete le motif
// l est la demi-longueur d'un motif
// h est la hauteur d'un motif
```

Exemple :

```
afficherFrise( 5, 6, 7);
```

```
*****      *****      *****      *****      *****
*   *   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *
*   *****      *****      *****      *****      *****
```