

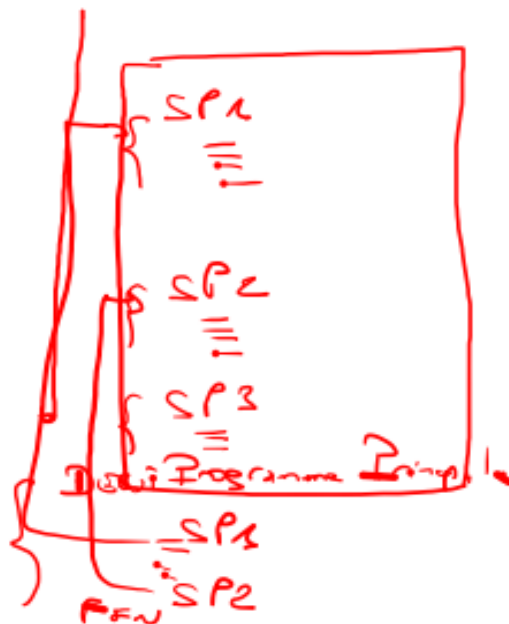
Sous Programme :

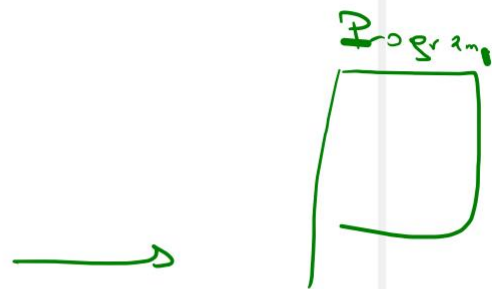
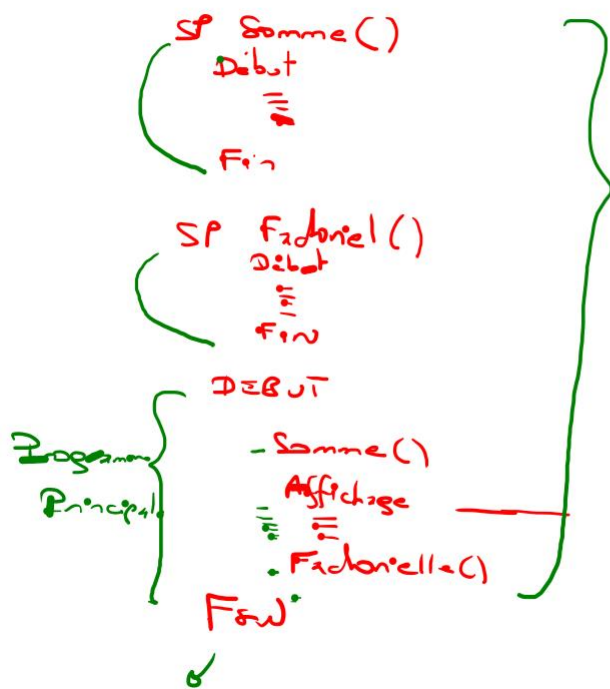
1. Organisation du Code, chaque Sous Programme est réservé pour une tâche (facile à maintenir)
2. Si un traitement est utilisé plus fois.

$$C_n^p = \frac{n!}{p!(n-p)!}$$

X

$$\begin{aligned} 1 &\equiv n! \\ 1 &\equiv p! \\ 1 &\equiv X! \end{aligned}$$

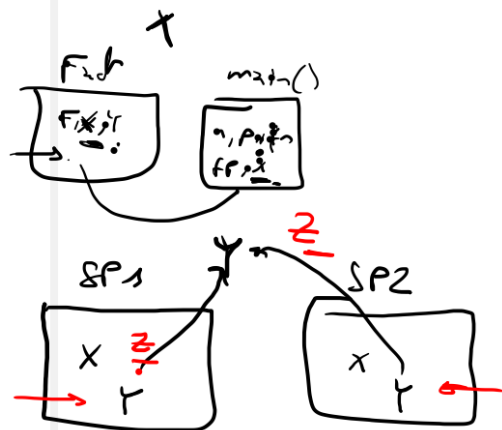




SP :
Procédure : exécute une tâche
Fonction : exécute une tâche
retourne une Valeur

Arguments
Fonction Fact (n : entier) —
Début
Var Fx : entier
F ← 1
Pour X ← 1 jusqu'à n faire
F ← F * X
Fpar
Retourner F
Fin
Var n, p, Fp, Fx : entier
Début
Lire(n, p)
Fp ← Fact(n)
Fx ← Fact(p) — Paramètre
x ← n - p
F ← Fact(x)
C ← Fp / (Fp * Fx)
Ecrire(C)
Fin

$$C_n^p = \frac{n!}{p!(n-p)!}$$



Déclaration
var z: entier

V. locale
V. globale

- Procédure Bonjour ()
Début

Ecrire ("Bonjour")
Ecrire (z)
Fin

- Fonction Bonjour ()
var x: entier
Début

Ecrire ("Bonjour")
x ← 10 + z
retourner x

Fin

Var A: entier

DEBUT

Bonjour()

A ← Bonjour()

z ← 100

FIN



void Bonjour ()
{

printf ("Bonjour\n");

}

int Bonjour ()

{

printf ("Bonjour\n");

x ← 10 + z;

} return x;

main()

{

Bonjour();

A = Bonjour();

}