

TD N° 1

Exercice 1

Donner le graphe de précédence pour ce code :

1. Read a, b, c
2. Delta = sqrt (b*b – 4.0 * a * c)
3. X1 = (-b + delta) /2.0/a
4. X2 = (-b – delta) /2.0/a

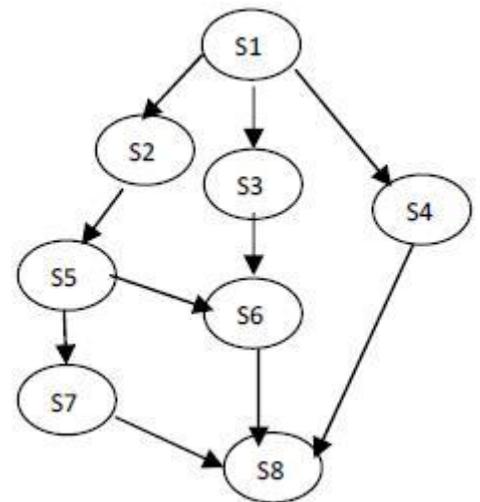
Exercice 2

1) On cherche à évaluer l'expression suivante

$$e := ((b-d) * (a+c) + (e*f)) / (a+c)$$

- a) Réaliser un découpage en tâches de cette expression sans l'ajout de variables intermédiaires.
- b) En vous servant de la définition de la condition de Bernstein, donner le graphe de précédence correspondant.

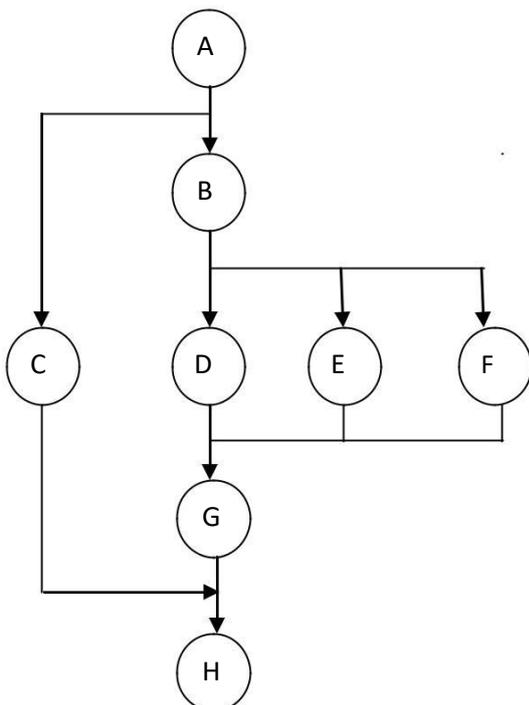
2) Soit le graphe de précédence suivant :



Donner le programme parallèle correspondant en utilisant les ParBegin/ParEnd.

Exercice 3

Utiliser ParBegin et ParEnd pour. Ecrire le code pour ce graphe :



Exercice 4

Soient les deux fonctions C suivantes :

<pre>int tab[10000] ; void fonction1() { int i ; for (i=0;i<500;i++) { tab[i]=i; } }</pre>	<pre>void fonction2() { int i ; for (i=500;i<10000;i++) { tab[i]=i; } }</pre>
--	--

Les cases du tableau tab sont initialisées à zéro. Donnez le contenu du tableau tab après la fin d'exécution des deux fonctions dans les trois cas suivants :

- Les deux fonctions s'exécutent dans deux processus différents (chaque fonction dans un seul processus chacune).
- Les deux fonctions s'exécutent dans deux threads du même processus (chaque fonction dans un seul thread chacune).
- Les deux fonctions s'exécutent dans deux threads de deux processus différents (chaque fonction dans un seul thread de chaque processus).