

Compte Rendu TP3

Question1

```
taches(Taches):-
    Taches = [](tache(3, [],m1, _),
    tache(8, [],m1, _),
    tache(8, [4,5], m1, _),
    tache(6, [],m2, _),
    tache(3, [1],m2, _),
    tache(4, [1,7], m1, _),
    tache(8, [3,5], m1, _),
    tache(6, [4],m2, _),
    tache(6, [6,7], m2, _),
    tache(6, [9,12], m2, _),
    tache(3, [1],m2, _),
    tache(6, [7,8], m2, _)).
```

Test

```
taches(Taches).
```

Question2

```
ecrit_taches(Taches):-
    dim(Taches,Dim),
    D is Dim[1],
    (for(I,1,D),
    param(Taches)
    do
        Var is Taches[I],
        writeln(Var)
    ).
```

Test

```
taches(Taches),ecrit_taches(Taches).
```

Question3

```
domaines(Taches,Fin):-
    dim(Taches,Dim),
    D is Dim[1],
    (for(I,1,D),
    param(Taches,Fin)
    do
        tache(_,_,M,D) is Taches[I],
        M::machine,
        D#>=0,
        D#<Fin
    ).
```

Test

```
taches(Taches),domaines(Taches,5).
```

Question4

```
getVarList(Taches,Fin,[Fin|Liste]):-
    dim(Taches,Dim),
    D is Dim[1],
    (for(I,1,D),
    param(Taches),
    fromto([],In,Out,Liste)
    do
        tache(_,_,Machine,Debut) is Taches[I],
        Out = [Debut|In]
    ).
```

Test

```
taches(Taches),domaines(Taches,5),getVarList(Taches,5,List).
```

Question5

```
solve(Taches,Fin):-
    taches(Taches),
    FinTotal = 0,
    domaines(Taches,Fin),
    precedences(Taches),
    conflicts(Taches),
    getVarList(Taches,Fin,Liste),
    labeling(Liste),
```

```
ecrit_taches(Taches).
```

Test

```
taches(Taches),solve(Taches,Fin).
```

Question6

```
precedences(Taches):-  
    dim(Taches,Dim),  
    D is Dim[1],  
    (for(I,1,D),  
    param(Taches)  
    do  
        tache(_, Noms,_, Debut) is Taches[I],  
        (foreach(Nom,Noms),  
        param(Taches,Debut)  
        do  
            tache(DureePrec, _, _, DebutPrec) is Taches[Nom],  
            Debut#>= DureePrec+DebutPrec  
        )  
    ).
```

Test

```
taches(Taches),solve(Taches,Fin).
```

Question7

```
conflicts(Taches):-  
    dim(Taches,Dim),  
    D is Dim[1],  
    (for(I,1,D),  
    param(Taches,D)  
    do  
        tache(Duree, _, Machine, Debut) is Taches[I],  
        I2 is I+1,  
        (for(J,I2,D),  
        param(Taches,Debut,Duree)  
        do  
            tache(Duree2, _, Machine2, Debut2) is Taches[J],  
            (Machine &\= Machine2) => ((Debut#>=Debut2+Duree2) or (Debut+Duree#=>Debut2))  
        )  
    ).
```

