



TP 2 – Structures conditionnelles

Exercice 1 : Écrire l'instruction suivante sans utiliser l'opérateur **and** .

```
if SommetTrefle(1) and TasNonVide(2)
then
  begin
    DeplacerSommet(1,2);
  end;
```

Sous quelle condition ce programme s'exécute-t-il correctement ? Écrire un programme qui vous permet de tester vos réponses.

Exercice 2 : Écrire l'instruction suivante sans utiliser l'opérateur **or** .

```
if SommetTrefle(1) or TasNonVide(2)
then
  begin
    DeplacerSommet(1,2);
  end;
```

Sous quelle condition ce programme s'exécute-t-il correctement ? Écrire un programme qui vous permet de tester vos réponses.

Exercice 3 : Écrire l'instruction suivante sans utiliser les opérateurs **and** et **not** .

```
if not (SommetTrefle(1)) and TasNonVide(2)
then
  begin
    DeplacerSommet(1,2);
  end
else
  begin
    DeplacerSommet(1,3);
  end;
```

Sous quelle condition ce programme s'exécute-t-il correctement ? Écrire un programme qui

vous permet de tester vos réponses.

Exercice 4 : Considérons la situation initiale où les tas sont initialisés par la séquence d'instructions suivante :

```
InitTas(1, '(T+K+P+C)[T+K+P+C]');  
InitTas(2, '(T+K+P+C)[T+K+P+C]');  
InitTas(3, '(T+K+P+C)[T+K+P+C]');  
InitTas(4, "");
```

Il faut déplacer sur le tas 4 la carte de plus grande valeur au sommet des trois premiers tas.

Question 4.1 : Expliquez pourquoi le programme suivant n'est pas correct pour résoudre ce problème. L'instruction `Superieur(i,j)` vaut `true` si la carte au sommet du tas i a une valeur plus grande que la carte au sommet du tas j , et vaut `false` sinon.

```
if Superieur(1,2) and Superieur(1,3)  
then  
  begin  
    DeplacerSommet(1,4);  
  end;  
if Superieur(2,1) and Superieur(2,3)  
then  
  begin  
    DeplacerSommet(2,4);  
  end;  
if Superieur(3,1) and Superieur(3,2)  
then  
  begin  
    DeplacerSommet(3,4);  
  end;
```

Question 4.2 : Proposez une solution correcte et testez-la.

Question 4.3 : Proposez une solution correcte, la tester, pour l'initialisation suivante des tas :

```
InitTas(1, '[T+K+P+C]');  
InitTas(2, '[T+K+P+C]');  
InitTas(3, '[T+K+P+C]');  
InitTas(4, "");
```