

TD 3

Présentation : Ce TD a pour but d'appréhender la notion de graphe de contrôle et les jeux de test.

Exercice 1

Soit la procédure P suivante :

```
1: ProcedureP(C1, C2 : BOOL)
2: BEGIN
3: if C1 then
4:   WRITE(C1) ;
5: end if
6: if C2 then
7:   WRITE(C2) ;
8: end if
9: END
```

- Construire le graphe de contrôle associée à cette procédure.
- Calculer le nombre cyclomatique de cette procédure.
- Donner les jeux minimaux de valeurs permettant de couvrir :
 1. les instructions
 2. les conditions
 3. les branches
 4. les chemins

Exercice 2

Soit la procédure suivante :

```
1: PROCEDURE P (C1, C2: BOOL)
2: BEGIN
3: if C1 OR C2 then
4:   WRITE("C1 C2")
5: end if
6: END
```

- Construire le graphe de contrôle associé à cette procédure.
- Calculer le nombre cyclomatique de cette procédure.
- Donner les jeux minimaux de valeurs permettant de couvrir :
 1. les instructions.
 2. les conditions (analyser les jeux de tests (T, F) et (F, T)).
 3. les branches.
 4. les chemins.

Exercice 3

Soit l'algorithme d'Euclide qui calcule le pgcd (plus grand commun diviseur) de deux nombres :

```
1: BEGIN
2: READ(x) ;
3: READ(y) ;
4: while not(x=y) do
5:   if x>y then
6:     x:=x-y ;
7:   else
8:     y:=y-x ;
9:   end if
10: end while
11: pgcd:=x ;
12: END
```

- Construire le graphe de contrôle associé à cette procédure.
- Calculer le nombre cyclomatique de cette procédure.
- Donner les jeux minimaux de valeurs permettant de couvrir :
 1. les instructions.
 2. les conditions.
 3. les branches.
 4. les chemins.
 5. les 2-chemins.

Exercice 4

Soit le programme suivant :

```
1: BEGIN
2: READ(x) ;
3: READ(y) ;
4: z:=0 ;
5: signe:=1 ;
6: if x<0 then
7:   signe:=-1 ;
8:   x:=-x ;
9: end if
10: if y<0 then
11:   signe:=-1 ;
12:   y:=-y ;
13: end if
14: while x≥y do
15:   x:=x-y ;
16:   z:=z+1 ;
17: end while
18: z:=signe*z;
19: END
```

- Déterminer les entrées et les sorties du programme.
- Construire le graphe de contrôle associé à cette procédure.
- Calculer le nombre cyclomatique de cette procédure.

- Donner les jeux minimaux de test permettant de couvrir :
 1. les instructions.
 2. les conditions.
 3. les branches.
 4. les 2-chemins.

Exercice 5

Soit le programme suivant :

```

1: Procedure search(desiredelement : ELEMENT, table: ELEMENT[], numberofitems: INTEGER)
2: VAR
3: found: boolean;
4: counter: integer ;
5: BEGIN
6: found:=false ;
7: if numberofitems >0 then
8:   counter:=1 ;
9:   while (not found) and (counter< numberofitems) do
10:    if table(counter)=desiredelement then
11:      found:=true ;
12:      counter:=counter+1 ;
13:    end if
14:  end while
15: end if
16: if found then
17:   write("the desired element exists") ;
18: else
19:   write("the desired element does not exist") ;
20: end if
21: END

```

- Déterminer les entrées et les sorties du programme.
- Construire le graphe de contrôle associé à cette procédure.
- Calculer le nombre cyclomatique de cette procédure.
- Donner les jeux minimaux de test permettant de couvrir :
 1. les instructions.
 2. les conditions.
 3. les branches.
 4. les chemins.