PROBLEME1 - SOLUTION1 : Utilisation de **ARRAY\_OF**

**ALGORITHM** Linear\_Data\_ARRAY\_OF

**VAR**

Ensemble1 : ARRAY\_OF INTEGER[50];

Ensemble2 : ARRAY\_OF INTEGER[50];

Ens-distinct : ARRAY\_OF INTEGER[100];

taille1, taille2, i, j, c, k ,x: INTEGER;

taille3 : INTEGER:=0;

existe : BOOLEAN;

**BEGIN**

**REPEAT**

**WRITE**("Entrer la taille de 1er ensemble"); **READ**(taille1);

**WRITE**("Entrer la taille de 2eme ensemble"); **READ**(taille2);

**UNTIL** (taille1<=50 **AND** taille2<=50);

**FOR** i **FROM** 0 **TO** taille1-1 **STEP** 1 **DO**

**WRITE**("Entrer un entier");

**READ**(Ensemble1[i]);

**END\_FOR**

**FOR** j **FROM** 0 **TO** taille2-1 **STEP** 1 **DO**

**WRITE**("Entrer un entier");

**READ**(Ensemble2[j]);

**END\_FOR**

**FOR** c **FROM** 0 **TO** taille1-1 **STEP** 1 **DO**

existe := false;

**FOR** k **FROM** 0 **TO** taille2-1 **STEP** 1 **DO**

IF(Ensemble1[c]=Ensemble2[k]) THEN existe := true;

**END\_FOR**

**IF**(existe=false) **THEN** Ens-distinct[taille3] := Ensemble1[c];

**END\_IF**

taille3 := taille3+1;

**END\_FOR**

**FOR** x **FROM** 0 **TO** taille3-1 **STEP** 1 **DO**

**WRITE**(Ens-distinct[x]);

**END\_FOR**

**END**

# PROBLEME1 - SOLUTION2 : Utilisation de Linked List

**ALGORITHM** Linear\_Data\_Linked\_List

**VAR**

LIST : **STRUCT**

data **: INTEGER**;

next : LIST;

**END\_STRUCT**

first1 : **^**List:=**NULL**;

first2 : **^**List:=**NULL**;

first3 : **^**List:=**NULL**;

Ensemble1 : **^**List:=**NULL**;

Ensemble2 : **^**List:=**NULL**;

Ens-distinct : **^**List:=**NULL**;

existe : **BOOLEAN;**

i, j, k, x, y : **INTEGER;**

**BEGIN**

**WHILE**(i<10) **DO**

**READ**(x);

Ensemble1 := create(LIST);

Ensemble1^.data :=x;

Ensemble1^.next := first1;

first1 := Ensemble1;

i:=i+1;

**END\_WHILE**

**WHILE**(j<10) **DO**

**READ**(y);

Ensemble2 := create(LIST);

Ensemble2^.data:=y;

Ensemble2^.next := first2;

first2 := Ensemble2;

j:=j+1;

**END\_WHILE**

**WHILE(**Ensemble1^**<>NULL) DO**

existe **:= false:**

**WHILE(**Ensemble2^**<>NULL) DO**

**IF(**Ensemble1^.data = Ensemble2^.data**) THEN** existe:=true;

**END\_IF**

**END\_WHILE**

**IF(**existe = false**) THEN**

Ens-distinct := create(LIST);

Ens-distinct^.data:= Ensemble1^.data;

Ens-distinct^.next := first3;

first3 := Ens-distinct;

**END\_IF**

**WHILE(**Ens-distinct^**<>NULL) DO**

**WRITE(**Ens-distinct^.data);

Ens-distinct^.next;

**END\_WHILE**

**END**

PROBLEME2 - SOLUTION1 : Utilisation de **ARRAY\_OF**

**ALGORITHM** Linear\_Data\_ARRAY\_OF

**VAR**

Ensemble1 : ARRAY\_OF INTEGER[50];

Ensemble2 : ARRAY\_OF INTEGER[50];

Ens-distinct : ARRAY\_OF INTEGER[100];

taille1, taille2, i, j, c, k ,x: INTEGER;

taille3, somme : INTEGER:=0;

existe : BOOLEAN;

**BEGIN**

**REPEAT**

**WRITE**("Entrer la taille de 1er ensemble"); **READ**(taille1);

**WRITE**("Entrer la taille de 2eme ensemble"); **READ**(taille2);

**UNTIL** (taille1<=50 **AND** taille2<=50);

**FOR** i **FROM** 0 **TO** taille1-1 **STEP** 1 **DO**

**WRITE**("Entrer un entier");

**READ**(Ensemble1[i]);

**END\_FOR**

**FOR** j **FROM** 0 **TO** taille2-1 **STEP** 1 **DO**

**WRITE**("Entrer un entier");

**READ**(Ensemble2[j]);

**END\_FOR**

**FOR** c **FROM** 0 **TO** taille1-1 **STEP** 1 **DO**

existe := false;

**FOR** k **FROM** 0 **TO** taille2-1 **STEP** 1 **DO**

IF(Ensemble1[c]=Ensemble2[k]) THEN existe := true;

**END\_FOR**

**IF**(existe=true) **THEN** Ens-distinct[taille3] := Ensemble1[c];

somme := somme+ Ensemble1[c];

**END\_IF**

taille3 := taille3+1;

**END\_FOR**

**WRITE(**"Somme des éléments qui se chevauchent:", somme\*2);

**FOR** x **FROM** 0 **TO** taille3-1 **STEP** 1 **DO**

**WRITE**(Ens-distinct[x]);

**END\_FOR**

**END**

# PROBLEME2 - SOLUTION2 : Utilisation de Linked List

**ALGORITHM** Linear\_Data\_Linked\_List

**VAR**

LIST : **STRUCT**

data **: INTEGER**;

next : LIST;

**END\_STRUCT**

first1 : **^**List:=**NULL**;

first2 : **^**List:=**NULL**;

first3 : **^**List:=**NULL**;

Ensemble1 : **^**List:=**NULL**;

Ensemble2 : **^**List:=**NULL**;

Ens-distinct : **^**List:=**NULL**;

existe : **BOOLEAN;**

i, j, k, x, y : **INTEGER;**

somme **: INTEGER :=0;**

**BEGIN**

**WHILE**(i<10) **DO**

**READ**(x);

Ensemble1 := create(LIST);

Ensemble1^.data:=x;

Ensemble1^.next := first1;

first1 := Ensemble1;

i:=i+1;

**END\_WHILE**

**WHILE**(j<10) **DO**

**READ**(y);

Ensemble2 := create(LIST);

Ensemble2^.data:=y;

Ensemble2^.next := first2;

first2 := Ensemble2;

j:=j+1;

**END\_WHILE**

**WHILE(**Ensemble1^**<>NULL) DO**

existe **:= false:**

**WHILE(**Ensemble2^**<>NULL) DO**

**IF(**Ensemble1^.data = Ensemble2^.data**) THEN** existe:=true;

**END\_IF**

**END\_WHILE**

**IF(**existe = true**) THEN**

Ens-distinct := create(LIST);

Ens-distinct^.data := Ensemble1^.data;

Ens-distinct^.next := first3;

first3 := Ens-distinct;

somme : = somme + Ensemble1^.data;

**END\_IF**

**WRITE("**Somme des éléments qui se chevauchent:", somme);

**WHILE(**Ens-distinct^**<>NULL) DO**

**WRITE(**Ens-distinct^.data);

Ens-distinct^.next;

**END\_WHILE**

**END**