

2^{ème} Epreuve de Moyenne Durée

Exercice 1 : (10 points)

Soit le programme suivant :

```
Programme principal
Begin
  Integer I, L, Array V1[1:L ; -1: 2*L ; 0:L+2];
  L1: Procedure Alpha (x1); Integer x1;
    Begin
      Integer J;
      L2 : Procedure Beta (x2); Integer x2;
        Begin
          Integer S, P, Array V2[S:P];
          Begin
            Integer X, Y;
            Read (X,Y);  S:=X+Y;  P:=X*Y;
            L3 : P:= (P * (-S/2)) + (X - X/Y);
          End ;
          L4: Begin
            Integer Array T [0 : 2*L];
            For I:= 0 To 2*L Do
              L5 : T[I]:= V1[I,J,I] * V2[I];
            L6: End;
          L7 : End;
          If (x1 < 20) Then Beta (x1*x1);
        End;
      L8 : Procedure Gamma (x3); Integer x3;
        Begin
          Integer Array V3 [-L : L], Val;
          L9 : V3[I]:= Val + V1[1,-1,0];
        End;
      Read (I);
      If ( I > 10) Then Alpha (I)
        Else Gamma (I);
    End;
End;
```

1. Donner les différents états de la pile de données aux étiquettes L1,L2,L3, ... jusqu'à L9.
2. Calculer les adresses absolues des variables de L5 sachant que les éléments de tous les tableaux sont rangés colonne par colonne.
3. Générer le code de l'instruction à L3 en utilisant les quadruplés et la machine vue en cours.

Exercice 2 : (10 points)

Soit l'instruction suivante :

$\text{Id} := \text{OR} (\text{cond}_1, \text{cond}_2, \dots, \text{cond}_n) \text{Exp}_1;$
 $\text{OTHERWISE Exp}_2;$

Sémantique:

L'identificateur Id reçoit la valeur de l'expression1 si une des conditions est vérifiée. Autrement, si toutes les conditions sont fausses, Id reçoit la valeur de l'expression2. On suppose que $n > 1$.

1. Donner la grammaire syntaxique.
2. Donner le schéma de traduction sous forme de ~~triples~~, dans le cas d'une analyse descendante.

gnadimples