

## 1 Exercice (d'après examen de Polytech, 2005/2006)

```

proc P1() {
  int x1 ;
  proc Q(int a) {
    x1 := a
  }
  proc P2 () {
    int x2
    proc P3 (int b) {
      if b>1 then
        P3(b-1) ; x1:=x2+1
      else
        Q(x1)
      fi
    } /* de P3 */
    x2:=1 ;
    P3(2)
  } /* de P2 */
  x1:=0 ;
  P2()
} /* de P1 */

```

**Q1.** Dessinez le contenu de la pile lors de l'exécution de la procédure Q. On précisera les liens statiques et dynamiques des procédures appelées.

**Q2.** Donnez le code assembleur de la procédure P3.

## 2 Correction

**Q1.** Pile :

**Q2.** Code de  $P_3$  :

```

prologue(0)
-- if b then
LD R1, [FP+12]           -- parametre b à l'adr. FP+12
CMP R1, 1
BLE sinon
-- P3(b-1)
ADD R2, R1, -1           -- R2 contient b-1
empiler(R2)              -- on empile le paramètre b-1
LD R3, [FP+8]           -- LS(P3) à l'adr. FP+12
empiler(R3)              -- on empile LS(P3)
CALL P3
ADD SP, SP, 8           -- on "dépille" LS(P3) et b-1

```

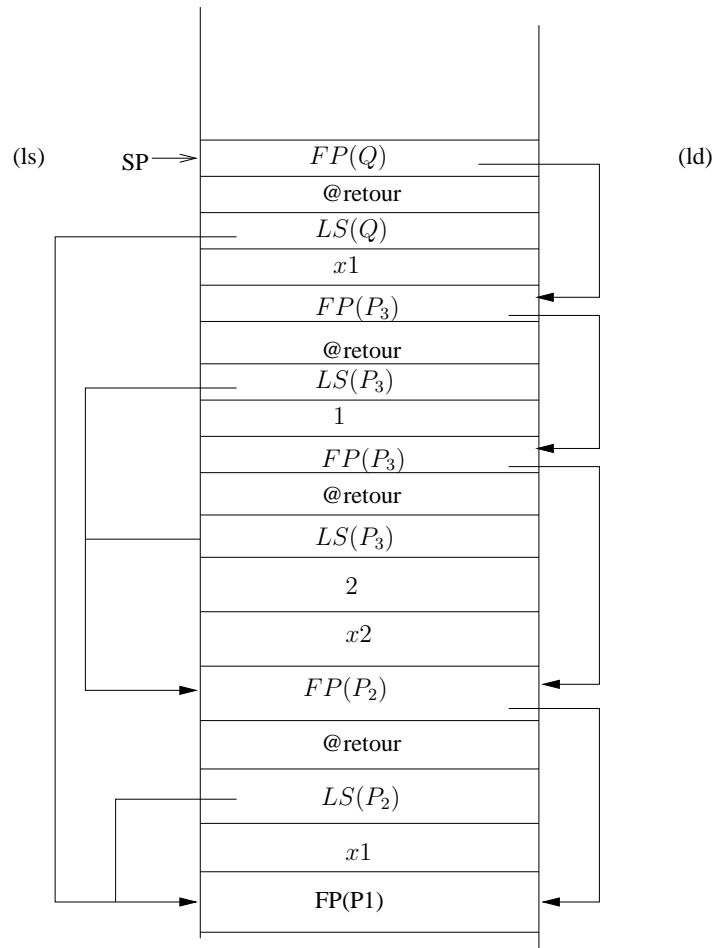


FIG. 1 – Pile à l'exécution de Q

```

-- x1:=x2+1
LD R4, [FP+8]
LD R5, [R4-4]
ADD R5, R5, 1
LD R6, [R4+8]
ST R5, [R6-4]
BA fsi
sinon:
-- Q(x1)
LD R7, [FP+8]
LD R8, [R7+8]
LD R9, [R8-4]
empiler(R9)
empiler(R8)
CALL Q
ADD, SP, SP, 8
fsi:
epilogue()
RET
-- env(x2)=env(P2) => une indirection sur LS(P3)
-- R5 contient x2
-- R5 contient x2+1
-- env(x1)=env(P1) => deux indirections sur LS(P3)
-- variable x1 à l'adr. R6-4
-- env(x1)=env(P1) => deux indirections sur LS(P3)
-- variable x1 à l'adr. R8-4
-- on empile le paramètre x1
-- on empile LS(Q)
-- on "dépile" LS(Q) et x1

```