Series 1

Exercise 1

We consider the two following extensions to the \texttt{while} language:

- A “conditional arithmetic expression” like in C or Java, with the following syntax
  \[ b \ ? \ e_1 : e_2 \], and those value is \( e_1 \) when \( b \) is true and \( e_2 \) otherwise.
- A “for” statement: \texttt{for x in e1 . . . e2 do S}

Give the corresponding code generation functions.

Exercise 2

Extend the code generation functions of the \texttt{while} language to deal with static \texttt{arrays}, according to the following syntax:

\[
\begin{align*}
D_V & ::= \cdots \mid \textit{var} x : \text{array} [n] \text{of Int} \\
A & ::= \cdots \mid x[A] \\
S & ::= \cdots \mid x[A] := A
\end{align*}
\]

Using these functions, give the code obtained for the following example:

\begin{verbatim}
var i : Integer ;
var T : array [10] of Integer ;
i := 0 ;
T[0] := 0 ;
while i < n-1 do T[i+1] := T[i] + i ;
\end{verbatim}

Generalize to bi-dimensional arrays, following the same syntax (e.g, \texttt{array [l][c] of Int}).

Exercise 3

Extend the code generation functions of the \texttt{while} language to deal with \texttt{result} and \texttt{reference} parameter modes. Using these functions, give the code obtained for the following example:

\begin{verbatim}
var x1 : Integer ;
procedure p1 (a : value Integer) is
begin
    var x2 : Integer ;
x2 := a+2 ;
end ;
\end{verbatim}
procedure p2 (a : ref Integer) is a := a+1;

procedure p3 (a : result Integer) is a := 1;

x1 := 42;
call p1(x1+1);
call p2(x1);
call p3(x1);

Exercise 4

procedure P(void);
  var x;

  procedure P1 (a)
  var x1;

  procedure P2(b, c);
  var x2;
  begin { P2 }
    x2 := c;
    x := x1 + x2 + b; /* instruction (2) */
  end; { P2 }

  begin { P1 }
    x1 := a;
    P2 (x+1, a); /* instruction (1) */
    x := 2;
  end; { P1 }

begin { P }
  x := 0;
  P1 (5);
end { P }

1. Give the execution stack during execution of procedure P2.
2. Give the code obtained for instructions labelled (1) and (2).

Exercise 5

We consider the following program P.
1. Give the “call tree” corresponding to the execution of P
2. Give the execution stack corresponding to the execution of X
3. Give the code produced for procedures P3 and R3.
void P() {
    int x1;
    void P1() { /* nesting level 1*/
        int x2;
        int y2;
        int z2;

        int Y(int x) {
            printf("%d\n", x+1);
            return (x+1);
        } /* Y */

        void R2(int b3, int p(int)) { /*nesting level 2*/
            int y3;
            void R3(int p(int)) {
                y2 = 2;
                x2 = y2+x1+b3;
                y3 = 1+x2;
                b3 = p(x1);
            } /* R3 */
            R3(p);
        } /* R2 */

        void P2(int a, int p(int)) { /*nesting level 2*/
            int x3;
            void P3(int p(int)) {
                x3 = p(x1);
                R2(x3,Y);
            } /* P3 */
            x2 = 1+a;
            P3(p);
        } /* P2 */

        int X(int x) {
            printf("%d\n", x);
            return x;
        } /* X */

        void Q() { /*nesting level 2*/
            int z3;
            y2 = 2;
            z2 = 3;
            z3 = x2 + y2;
            P2(z3,X);
        } /* Q */

        x1 = 11;
        Q();
    } /* P1 */
}
} /* P */