問題 I

以下は，言語 $\mathcal{W}$ の操作的意味にしたがって ML (Standard ML) で記述したインタプリタの抜粋であり，red_cexp は，プログラムの抽象構文木（型cexp の要素）と初期状態を受け取り，終了状態を返す関数である．下線部 (_____) に適切な式を埋めよ．

```
datatype cexp = Skip | Assign of varname * aexp | Seq of cexp * cexp
              | If of bexp * cexp * cexp | While of bexp * cexp
(* 命令を評価するための関数 *)
fun eval_inst_cexp (ci: cexp) (st: state): cexp*state =
    case ci of
      Assign(x, a) =>
        if reducible_aexp(a) then
          let val a' = red_aexp_onestep a st in (Assign(x, __a'___), st) end
        else let val st' = update_state st x a  in (Skip, st') end
   | Seq(Skip, c1) => (_c1__, st)
   | If(b, c1, c2) =>
        if reducible_bexp(b) then
          let val b' = red_bexp_onestep b st in (If(b',c1,c2), st) end
        else case b of
             True => (__c1___, st)  | False => (c2, st)
   | While(b, c1) =>
        (If(__b___, Seq(c1, ci), Skip), st)
(* c が簡約可能か否かを返す関数 *)
fun reducible_cexp c = c!=Skip
(* プログラム c を文脈と命令に分解する関数 *)
fun split_cexp c = ...
(* 文脈 e とプログラム c を受け取り， e[c] を返す関数 *)
fun comb_cexp e c =
  case e of CChole => c | CCseq(e1, c1) => Seq(comb_cexp e1 c, c1)
(* 1 ステップ評価する関数 *)
fun red_cexp_onestep (c:cexp) (st:state) =
  let val (cc, i) = split_cexp c in let val (c', st') = eval_inst_cexp i st in
    (comb_cexp cc c', st')
  end end
(* c を評価し，終了状態を返す関数 *)
fun red_cexp c st =
   if reducible_cexp c then
      let val (c',st') = red_cexp_onestep c st
      in red_cexp c' st' end
   else __st____
```

問題 II

以下のプログラムを下の命令セットからなる言語に変換せよ.

$$\textbf{while not}(X = Y)\ \textbf{do if}\ X \leq Y\ \textbf{then}\ Y := Y - X\ \textbf{else}\ X := X - Y$$

命令	意味
$r \leftarrow v$	レジスタ r に値 v を格納
$r \leftarrow X$	変数 X の値をレジスタ r に格納
$X \leftarrow v$	変数 X に値 v を格納
$r \leftarrow v_1 + v_2$	v_1 と v_2 の和をレジスタ r に格納
$r \leftarrow v_1 - v_2$	v_1 と v_2 の差をレジスタ r に格納
$r \leftarrow \texttt{LTEQ}(v_1, v_2)$	比較 $v_1 \leq v_2$ の結果をレジスタ r に格納
$r \leftarrow v_1 = v_2$	比較 $v_1 = v_2$ の結果をレジスタ r に格納
$r \leftarrow \texttt{NOT}\ v$	v の否定をレジスタ r に格納
jmp L	ラベル L にジャンプ
jif_false L	直前の計算結果が偽ならばラベル L にジャンプ
L:	ラベル

注： v は整数またはレジスタ名

L_1 :
$r_0 \leftarrow X$
$r_1 \leftarrow Y$
$r_2 \leftarrow r_0 = r_1$
$r_3 \leftarrow \texttt{NOT}\ r_2$
$\texttt{jif\_false}\ L_3$
$r_4 \leftarrow \texttt{LTEQ}(r_0, r_1)$
$\texttt{jif\_false}\ L_2$
$r_5 \leftarrow r_1 - r_0$
$Y \leftarrow r_5$
$\texttt{jmp}\ L_0$
L_2 :
$r_5 \leftarrow r_0 - r_1$
$X \leftarrow r_5$
$\texttt{jmp}\ L_0$
L_3 :

問題 III （できる人のみ． ボーナス点あり．）

言語 $\mathcal{W}$ のコンパイラを作成せよ．（入力は抽象構文木でよく，出力は問題 II と同程度の命令セットの命令列とする．http://www.kb.ecei.tohoku.ac.jp/~koba/class/soft-eng/compiler.zip に未完成のコードがあるので、それを元にしてもよい．）提出は，メールでsoft-eng@kb.ecei.tohoku.ac.jp まで。