Subtask 4: Implement the code generation for control structures (see Slides 15-21 in the notes on "10-Intermediate Code Generation): able to test "fac.cm" and "booltest.cm".

Backpatching Example

```
* Standard prelude
0:  LD 6, 0(0)
1:  LDA 5, 0(6)
2:  ST 0, 0(0)
* Jump around i/o routines
* Code for input routine
4:  ST 0, -1(5)      store return
5:  IN 0, 0, 0       input
6:  LD 7, -1(5)      return to caller
* Code for output routine
7:  ST 0, -1(5)      store return
8:  LD 0, -2(5)      load output value
9:  OUT 0, 0, 0      output
10: LD 7, -1(5)      return to caller
3:  LDA 7, 7(7)      jump around i/o code
11:
```

```
/* code for backpatching */
...
int savedLoc = emitSkip(1);   //3
/*
   code for the i/o routines
*/
int savedLoc2 = emitSkip(0);   // 11
emitBackup( savedLoc );
emitRM_Abs( "LDA", pc, savedLoc2, "" );
emitRestore();
...
```


Three-Address vs Assembly Code

```
read x
t1 = x > 0
if_false t1 goto L1
fact = 1
label L2
t2 = fact * x
fact = t2
t3 = x - 1
x = t3
t4 = x == 0
if_false t4 goto L2
write fact
label L1
halt
```

```
0:    IN   0, 0, 0
1:    JLE  0, 6(7)
2:    LDC 1, 1, 0
3:    LDC 2, 1, 0
4:    MUL 1, 1, 0
5:    SUB 0, 0, 2
6:    JNE 0, -3(7)
7:    OUT 1, 0, 0
8:    HALT 0, 0, 0
```

```
int saved = emitSkip(0);

emitRM_Abs("LDA", pc, saved,"");
```

(Compared with 70 instructions "fac.tm")
Backpatching for forward jumps

Code Generation for Control Stmts

stmt -> if-stmt | while-stmt | **break** | **other**
if-stmt -> **if** (exp) stmt | **if** (exp) stmt **else** stmt
while-stmt -> **while** (exp) stmt
exp -> **true** | **false**

e.g., if (true) while (true) if (false) break else other

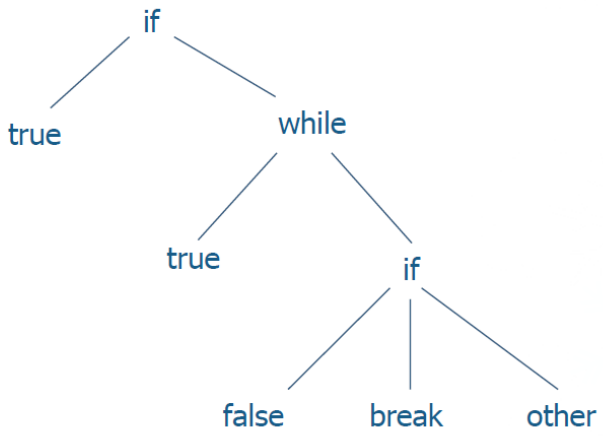
```
if_false true goto L1

label L2
if_false true goto L3

if_false false goto L4
goto L3
goto L5
label L4
other
label L5

goto L2
label L3

label L1
```



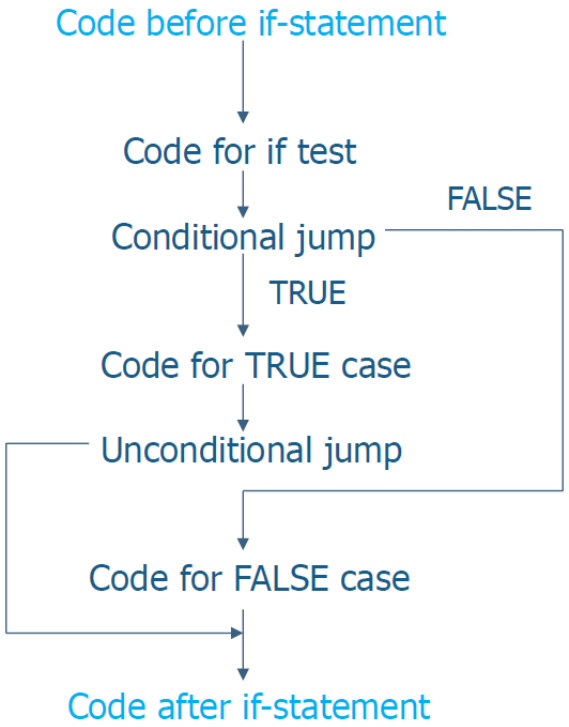

```
void genCode( Exp tree, String label ) {
   String codestr = "";
   String lab1, lab2;
   if( tree != null ) {
      if( tree instanceof IntExp ) {
        // do nothing
      } else if( tree instanceof IfExp ) {
        // refer to the related fragment
      } else if( tree instanceof WhileExp ) {
        // refer to the related fragment
      } else if( tree instanceof BreakExp ) {
        codestr += "goto  " + label;
        emitCode( codestr );
      } else if( tree instanceof OtherExp ) {
          emitCode( "Other" );
      } else
   }
}
```

```
// code fragment for WhileExp
lab1 = genLabel();
codestr += "label" + lab1;
emitCode( codestr );
genCode( tree.test, label );
lab2 = genLabel();
if( tree.test .value == 0 )
   codestr += "if_false false goto " + lab2;
else
  codestr += "if_false true goto " + lab2;
emitCode( codestr );
genCode( tree.body, lab2 );
codestr += "goto " + lab1;
emitCode( codestr );
codestr += "label " + lab2;
emitCode( codestr );
```

```
// code fragment for IfExp
genCode( tree.test, label );
lab1 = genLabel();
if( tree.test.value == 0 )
    codestr += “if_false false goto ” + lab1;
else
    codestr += “if_false true goto ” + lab1;
emitCode( codestr );
genCode( tree.then, label );
if( tree.else != null ) {
    lab2 = genLabel();
    codestr += “goto ” + lab2;
    emitCode( codestr );
}
```

```
// continued from left
codestr += “label ” + lab1;
emitCode( codestr );
if( tree.else != null ) {
    genCode( tree.else, label );
    codestr += “label ” + lab2;
    emitCode( codestr );
}
```

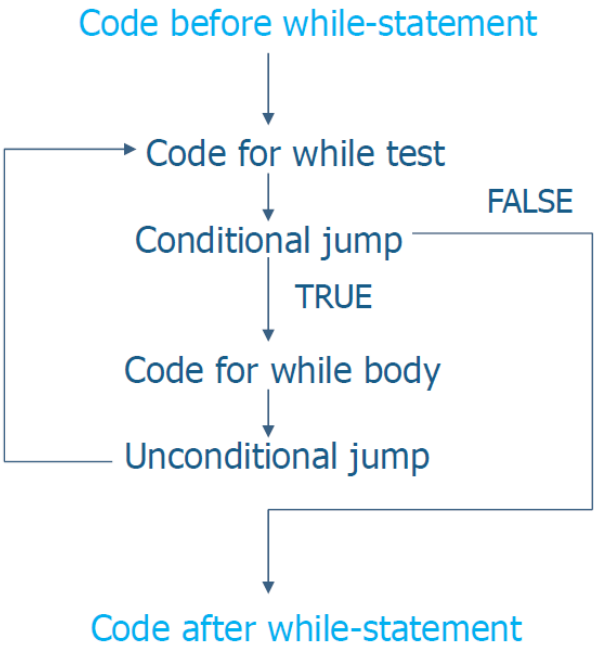
Code for If-statements



if (E) S1 else S2

```
<code to evaluate E to t1>
if_false t1 goto L1
<code for S1>
goto L2
label L1
<code for S2>
label L2
```

Code for While-statements



while (E) S

label L1
<code to evaluate E to t1>
if_false t1 goto L2
<code for S>
goto L1
label L2