COMP.2030 Homework 2 Due: 9/18 (Mon) 11:59 PM

Consider an array, X, containing only nonnegative integers. The last value in X is set to -1 to mark the end of the array.

For simplicity, suppose that the search key, key, is global. A linear search program can be written as on the left. The same logic can be written with ‘goto’ as on the right, called pseudo-C code.

void main() {

int i = 0;

int found = 0;

rept:

if (found) goto done;

if (X[i] == -1) goto done;

if (X[i] != key) goto else;

found = 1;

goto rept;

else: i++;

goto rept;

done:

if (!found) goto else2;

printf(%d, i);

goto exit;

else2:

printf(“-1”);

exit:

return;

}

void main() {

int i = 0;

int found = 0;

while (!found && X[i] != -1){

if (X[i] == key) found = 1;

else

i++;

}

if (found)

printf(%d, i-1);

else

printf(“-1”);

}

Write the pseudo-C code with registers and memory accesses. For a memory access, use the notation of ‘MEM[ ]’ to indicate the memory access with the memory address inside the bracket. For a Boolean variable, use 0 for false and 1 for true.

An ‘if’ statement in pseudo-C has to be written in two Assembly instructions of ‘comp(,)’ for a comparison followed by an if( ) with the condition (such as EQ or LT) inside the parenthesis combined with a goto. For example, if (x < y) goto skip would be written in assembly as the two instructions

comp(x, y);

if (LT) goto skip;

and ‘if (found) goto done’ would be written in assembly as the two instructions

comp(found, 1);

if (EQ) goto done;

What to submit:

Assembly language version of the pseudo-C code in docx or pdf.