Programming Lab 6E

Reversi (aka Othello)

Topics: Parameter passing, function return values, nested functions, preserving and restoring registers across function calls, calling C functions from assembly.

Prerequisite Reading: Chapters 1-6
Revised: February 3, 2021

Background[1]: Englishmen Lewis Waterman and John W. Mollett both claim to have invented the game of Reversi in 1883, each denouncing the other as a fraud ☺ . The game gained considerable popularity in England at the end of the nineteenth century. The goal for each player is to make pieces of their color constitute a majority of the pieces on the board at the end of the game, by turning over as many of their opponent's pieces as possible.

In this game (which is similar to the Othello version), you take turns playing against the computer. The game begins with four disks placed in a square in the middle of the grid, two red and two green, so that the same-colored disks are on a diagonal. Your pieces are green, while those of the computer are red. You must place a green piece so that there exists at least one straight (horizontal, vertical, or diagonal) line between the new piece and another green piece, with one or more contiguous red pieces between them. Legal placements are indicated in white. Touching a white spot replaces it and the surrounded red pieces by your green pieces. Pressing the blue pushbutton allows the computer to make its move. The game ends when no further moves are possible. The player with the most pieces wins.

Assignment: The main program may be compiled and executed without writing any assembly. However, your task is to create faster assembly language replacements for the two C functions shown below using their C versions to guide your implementation. The original C functions are defined as "weak", so that the linker will automatically replace them in the executable image by those you create in assembly; you do not need to remove the C version. Each function <u>must</u> use an `IT` block and must <u>not</u> use PUSH or POP instructions:
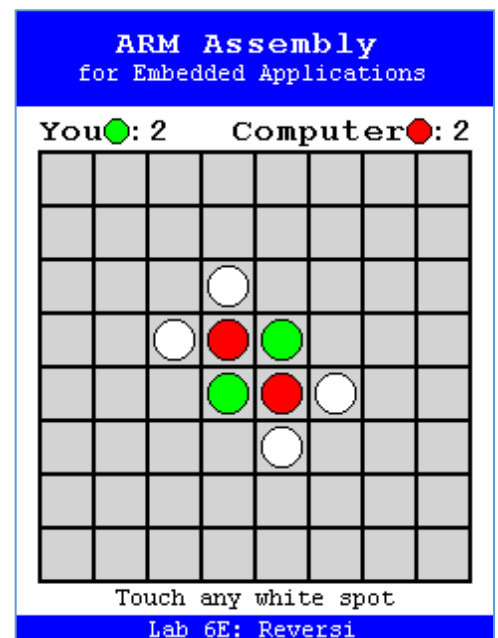
```
int Between(int min, int value, int max) ;
int Count(int cells[], int numb, int value) ;
```

Function `Between` should return true (1) if $min \leq value \leq max$, and false (0) otherwise. To receive full credit, use the following trick to reduce execution time by eliminating one of the comparisons and be prepared to explain why it is functionally equivalent:

$$(unsigned)\,(value - min) \leq (unsigned)\,(max - min).$$

Function `Count` should return an integer between 0 and $numb$ that is the number of elements in the $cells$ array whose content is equal to $value$. To receive full credit, use the auto-increment addressing mode to access the elements of the array.

Test your function using the main program. If you code works correctly, the display should look like the image on the right. Pressing the black pushbutton restarts the game.

[1] Adapted from https://en.wikipedia.org/wiki/Reversi