```
/**
 * CS 240 Connect Four: colcheck.s
 * Implement this function in assembly code.
 * Check if the board grid contains four connected pieces of type p
 * in a column, including the piece p just played at [row,col].
 *
 * Simpler, less efficient algorithm.
 */
/*
long checkcol(piece grid[6][7], long row, long col, piece p) {
  long i;
  for (i = 0; i <= row; i++) {
    if (grid[row - i][col] != p) {
      return 0;
    }
  }
  return i >= 4;
}
*/
    .text
    .align 8
    .globl checkcol

// long checkcol(piece grid[6][7], long row, long col, piece p);
checkcol:
    // grid = %rdi
    // row = %rsi
    // col = %rdx
    // p = %rcx
    // i = %r10
    // i = 0;
    movq $0, %r10
loopTop:
    // test i <= row ?  (a.k.a. if NOT i > row)
    cmpq %rsi, %r10
    jg loopEnd
    // get row - i   // REGISTER decision caller/callee
     // row
    movq %rsi, %r8
     // row - i
    subq %r10, %r8
      // scale row index by #cols
    imulq $7, %r8
    // add col
    addq %rdx, %r8
    // get grid[(row - i)*COLS + col]
     // element from grid
    movq (%rdi, %r8, 8),%r9
    // if (element != p)
    cmpq %r9, %rcx
    je endIf
    // return 0;
    movq $0, %rax
    jmp end
endIf:
     // i ++
    incq %r10
    jmp loopTop
loopEnd:
    // return i >= 4
    movq $0, %rax
    movq $1, %r11
    cmpq $4, %r10
    cmovge %r11, %rax
end:
    retq

```