CS 271 Computer Architecture and Assembly Language
Self-Check for Lecture #20 **EXAMPLE SOLUTION (one of several possible)**

Given the following partial data segment:

```
.data
loVal           DWORD       ?
hiVal           DWORD       ?
randVal         DWORD       ?


.code
main    PROC
        call  Randomize         ; from the Irvine library

;  Code to get loVal and hiVal from the user goes here.

        push  loVal
        push  hiVal
        push  OFFSET randVal
        call  nextRand

;  More main procedure code

        exit
main    ENDP
```

Write the *nextRand* procedure so that it satisfies the following header documentation, and is consistent with the call above. You may use appropriate Irvine library procedures. No fair using global variables. Note that used registers must be saved and restored.

```
; Procedure nextRand
; Procedure to get the nest random number in the range specified by the user.
; Receives parameters on the system stack (in the order pushed):
;       Lowest acceptable value (loVal)
;       Highest acceptable value (hiVal)
;       Address of return value
; Preconditions:  loVal < hiVal
; Registers used:  none
nextRand    PROC
      pushad                  ;save registers
      mov   ebp,esp           ;set stack frame pointer
      mov   eax,[ebp+40]      ;hiVal in eax
      sub   eax,[ebp+44]      ;subtract loVal
      inc   eax               ;  and add 1 to get the number of integers in range
      call  RandomRange       ;eax gets value in [0 .. range-1]
      add   eax,[ebp+44]      ;eax has value in [loVal .. hiVal]
      mov   edi,[ebp+36]      ;edi gets destination memory address
      mov   [edi],eax         ;send result to memory
      popad                   ;restore registers
      ret   12                ;return and clear activation record
nextRand    ENDP
```