

ASSEMBLY PROGRAMMING

Test sample

No aids can be used!

Timeframe: 50 minutes

Maximum: 50 points (expected: 50%)

Which statement is true about a code snippet written in assembly that implements a `while` structure?

- ☐ a. The code for the loop condition always precedes the code for the loop body statements.
- ☐ b. The code snippet does not contain unconditional control transfer.
- ☐ c. The code snippet contains one conditional and one unconditional control transfer statement.
- ☐ d. The first line of the code snippet contains the assembly instruction specified by the `while` mnemonic.

2 points

Three more similar questions...

Which registers will be changed/affected by the following Intel syntax x86-64 assembly instruction? Select the registers whose values can be different before and after the execution of the following instruction.

```
call abs
```

- ☐ a. `eax`
- ☐ b. `rsp`
- ☐ c. `rip`
- ☐ d. `edi`

2 points

Two more similar questions...

The following sequence of Intel syntax x86-64 assembly instructions is given. What will be the content of the "eax" register in signed fixed-point interpretation given in decimal notation, after the entire code is executed?

```
mov eax, 10011  
neg eax
```

Answer:

2 points

Three more similar questions...

Give the single Intel syntax x86-64 assembly instruction that works in the following way!

As a result of the instruction, the 16-bit value of the memory area referenced by the `var` label overwrites the 4-byte part of the `rax` register so that the value in the register is the same signed integer as the one in memory, only the representation length increases.

, PTR [var]

(Drag the selected item from the list below to the correct position. You can use each item in the list more than once, but some items are not necessary.)

3 points

Two more similar questions...

Create an x86-64 assembly code snippet that works similarly to the following C language code and produces similar machine code! Drag the 3 instructions selected from the list below to the appropriate place in the assembly code!

```
int abs1(int N){
    if(N<0) N=N*-1;
    return N;
}
```

abs1: mov eax, edi

end: ret

3 points

Two more similar questions...

Interpret the following Intel syntax x86-64 assembly code snippet! What is it for? What does it mean?

```
func: mov eax, 1
      cmp edi, 0
      jg .Lg
      cmp edi, 0
      je .Le
      dec eax
.Le:  dec eax
.Lg:  ret
```

- ☐ 1. The function should return the value that the mathematical sign function (`sign`) would return based on the integer value received as a parameter.
- ☐ 2. Returns the number of parameters of the function.
- ☐ 3. The function returns true if and only if the sum of its parameters is exactly 255.
- ☐ 4. The function returns the factorial of the integer given as a parameter.
- ☐ 5. The function returns the number of "1" bits in the integer received as a parameter.

5 points

One more similar question...