

ABDULLAH TAHIR

Bscs

13789

ESTR AMIN

MICROPROCESSOR AND ASSEMBLY LANGUAGE.

Q1) Use the following variables - - - ?

ANSWER :-

a - EAX = 0000ABCDH

b - EAX = FFFFABCDH

c - AX

d - AX

e - val2 = 3000

f - val =

g - AL = 50

h - AL = 40

i - AX = 3000h

j - AX = 1000h

k - EAX = 30000000h

l - EAX = error

Q2) Write down values . . . ?

ANSWER:-

$$a) CF = 1, SF = 0, ZF = 1, OF = 0$$

$$b) CF = 0, SF = 1, ZF = 0, OF = 1$$

$$c) CF = 0, SF = 1, ZF = 0, OF = 0$$

$$d) CF = 1, SF = 0, ZF = 0, OF = 1$$

Q3)2) Write an instruction - - - ?

ANSWER

mov eax, DWORD PTR listB

Q3)1) Use the following - - -

answer ::

a = EAX - 1

b = EAX - ~~32~~ 6

c = EAX - ~~16~~ 6

d = EAX - 2

e = EAX = 32

f = EAX = 64

g = EAX = 18

Q3)c) insert Label - - - ?

ANSWER :

list B LABEL DWORD

list B WORD 3 DUP (?) 2000h

data

mov eax, list B

Q4) Use the following data definition

ANSWER :-

a, AL = 10h

b, AL = 40h

c, AX = 003Bh

d, EDX = 3

e, EDX = 3

f, EAX = 2

Q5) implement the following Pseudocode.

```
(1) mov eax, var1  
    cmp eax, var2  
    jle L1  
    mov L1  
    mov var3, 10  
    mov var4, 30  
    jmp L2  
L1: mov var3, 15
```

```
(2) cmp val1, ecx  
    jna else  
    jna ecx, edx  
    jna else  
    mov a, 12  
    jmp next  
else mov b, 6  
next:
```

(3)

Qs.) What will be the final value?

ANS

mov eax, 0

mov ecx, 10; ^{loop} outer counter

L1: mov eax, 3

mov ecx, 5; inner loop counter

L2: add eax, 5

↳ loop L2; repeat inner loop
loop L1; repeat outer loop

Q6) Write a program - - - ?

386

model flat, std call

stack 4096

Exit Process PROTO, dwExitCode: DWORD

code

main PROC

mov eax, 3h

mov ebx, 8h

mov ecx, 1h

mov edx, 8h

add eax, ebx

add ecx, edx

sub eax, edx

INVOKE ExitProcess, 0

main ENDP

END main.

Q6)3) Write a program that _____?

386

.model flat, stdcall

.stack 4096

Exit process PROTO, dwExitCode : DWORD

data

array WORD 0, 2, 5, 9, 10

~~new~~ newArray DWORD LENGTHOF array DUP (?)

code

main PROC

move ecx, LENGTHOF array

MOV ESI, OFFSET array

MOV EDI, OFFSET newArray

L1:

MOV EAX, 0

MOV AX, [ESI]

MOV [EDI], EAX

ADD ESI, TYPE array

ADD EDI, TYPE ~~new~~ newArray

LOOP L1

INVOKE Exit process, 0

main ENDP

END main

Q6) Write a program ?

```
INCLUDE Irvine32.inc
```

```
.data
```

```
count DWORD ?
```

```
.code
```

```
main PROC
```

```
mov eax, 0 + (0 * 16)
```

```
mov ecx, 16
```

```
L1
```

```
mov count, ecx
```

```
push eax
```

```
mov ecx, 16
```

```
L2
```

```
call setTextColor
```

```
push eax
```

```
mov al, 'H'
```

```
call writechar
```

```
pop eax
```

```
inc eax
```

```
LoopL2
```

```
call crlf
```

```
pop eax
```

```
add eax, 16
```

```
mov ecx, count
```

```
LoopL1
```

```
call crlf
```

```
call waitMsg
```

```
exit
```

```
Main ENDP
```