

**ASSIGHMENT NO: 03**

**NAME: RAFI ULLAH**

**ID# 14283**

**DEPARTMENT: BS(CS)**

**SUBJECT: MICRO PROCESSOR & ASSEMBALY  
LANGUAGE**

**SEMESTER: 5<sup>th</sup>**

①

Q No 1

Using the value -35 write an integer literal in decimal, hexadecimal, octal and binary format that are constant with mask syntax.

Ans:

-35d, 0DH, 335o, 1101110110

Q No 3

write the real number  $-6.2 \times 10^4$  as a real number literal using MASK syntax.

Ans:

-62E+04

Q No 7 What is a calling convention and how is used in assembly language declaration?

Ans: A calling convention determines how parameters are passed to subroutines and how the stack is restored after the subroutine call.

(2)

**Q No. 18** Why might you use a symbolic constant rather than an integer literal in your code?

**Ans:** An integer literal such as 48 has no direct meaning to someone reading the program's source code. Instead, a symbolic constant such as `STUDENT_COUNT` can be assigned an integer value and is self-documenting.

**Q No. 25** How is source file different from a listing file?

**Ans** A source file is given as input to the assembler. A listing has additional text that will not assemble. It is a file that is created by the assembler and it is optionally generated.

3

3

**Q No 27**

How are data label and code label different?

Ans:

Ans: Data labels exist in the data segment as variable offsets. Code labels are in the code segment and are offset for transfer of control instructions. A code label is followed by a colon but a data label does not with a colon.

**Q No 5** Which statement halts the assembly language program?

Ans: The exit statement (indirectly) calls a predefined MS-Windows function that halts the program. The ENDP direct moves the end of the main procedure. The END main direct moves the last line of the program to be assembled. Data type



(4)

essential characteristics size  
in bit 8, 16, 32, 64.

**QNo. 8:**

Which type of files are produced by the assembler and linker?

Ans:

Ans: The main output produced by assembler on input assembly language source file is the translation of the file into an object file in (ELF) - ELF files produced by the assembler are relocatable files that hold code and/or data. They are input files for the linker.

**QNo. 9**

Which operating system component loads and execute programs?

Ans: Ans: A monolithic kernel runs all the operating system instruction in the same address space for speed. A microkernel run most process in user space for modularity. This central component of a computer system

5

is responsible for running or  
executing program

OR

Ans

The Loader.

Q No: 10

Declare an unsigned 16-bit integer  
variable named warray that uses  
three initializers.

Ans: warray WORD 10, 20, 30...

Q No: 17

Declare a string variable  
containing that name of your favourite  
color initialize it as a null-terminated  
string?

Ans: my\_color BYTE "blue" 0

Q No 12

Ans: array size = ( \$ - my array )



(6)

Q No: 20

Show how to calculate the number of element in the following array, and assign the value to a symbolic constant named Array Size:

my Array DWORD 30 DUP (?)

Ans. Array size = (\$-my Array) / Type DWORD

Q No: 20

Create an unsinged for a ~~16-bit~~ declaration for a 8-bit, 16-bit and 32-bit unsinged and signed integer?

Ans. signed - var 2 5 BYTE

\* 16-bit signed integer

var 1 DWORD

\* 8-bit signed integer

var 2 BYTE

\* 32-bit signed integer

7

Ans. SDWORD

Q No: 15

Ans. Little Endian:

The Least Significant byte (the "little end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order in the next bytes in memory.

\* In these definition, the data a 32-bit pattern is regarded as a 32-bit unsigned integer. The "most significant" byte is the one for the lowest power of two:  $2^{31}$   $2^{24}$ . The smallest power of two:  $2^7$   $2^0$

\* For example say that the 32-bit pattern  $0x12345698$  is stored at address  $0x00400000$



(8)

78	56	34	12
0	0	0	0
x	x	x	x
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

Q No 19

Show the order of individual bytes in memory (lowest to highest) for the following double word.

Ans: Variable: quad DWORD  
87654321h

True or false: The following is a valid data definition of statement.

var 1 BYTE 0Ah, 255

write an assembly program to compute the following expression

$$AL = BL + var1$$

var1 is an 8-bit variable.

(9)

**Q No: 23**

Difference b/w equal sign directive and equ directive.

Ans: **Equal Directive** sign :

The Equal Sign Directive associates a symbol name with an integer expression.

This syntax is

**name = Expression**

Expression is a 32-bit integer

- May be redefined
- Name is called a symbolic constant.

**EQU - directive**

Define a symbol as either an integer or text expression

- cannot be redefined

**Q No: 28**

Name the four basic parts of an assembly language instruction.

(10)

Ans: [Label:] mnemonic [operands]  
[: comment]

**Q No 29**

Show an example of a  
block comment?

Ans: COMMENT:  
First line comment  
Second line comment.

**Q No 30**

Why it is not good idea to use  
numeric address when writing  
instruction that access variable?

Ans: ~~too~~

Ans: ~~too~~ do not use numeric address  
(offsets) for variable because the  
addresses would change if new  
variable were inserted before the  
existing ones.



(11)

**Q No 2**

create a single integer expression that use all the operators calculate the value of the expression.

Ans:  $(5+1)(-2+3)*2 \text{ mod } 5 = 2$

~~Q No 1~~

**Q No: 6**

What type of argument must be passed to the Exit process procedure?

Ans: An integer, preferably 0.

**Q No 24**

Declare an array of 120 uninitialized unsigned double word value.

Ans:

Ans: `my Array DWORD 120 (?)`

(12)

QNo 15

Describe an array of byte and initialize it to the first 5 letters of the alphabets.

Ans: my Array BYTE 'A', 'B', 'C', 'D', 'E'

QNo 16

What type of arguments must be passed to the Exit process procedure?

Ans: mode small

• 326

• stack 100h

• data

Sunday = 0

Monday = 1

Tuesday = 2

Wednesday = 3

Thursday = 4

(13)

Friday = 5

Saturday = 6

Days - DB Sunday, Monday, Tuesday, Wednesday,  
Thursday, Friday, Saturday

• code

main;

max ax, @data

max ds, ax

; Just for test; print the first value.

max ah, 02h

max dl, days

add dl, 50h

int 21h

max ah, 4ch

int 21h

end main.



14

Q No: ~~28~~ 13

Find out if you can declare a variable of type DWORD and assign it a negative value.

Ans: V9U DWORD 12345678h; unsigned  
V9I3 DWORD -20 DUP(?); unsigned

Q No: ~~29~~ 4

Discuss the following MASK directive

Ans:

Stack directive:

The stack directive tells how many bytes of memory to reserve for the runtime stack. 4096 happens to correspond to the size of a memory page in the processor system for managing memory.

(15)

## MODEL :

This tells the assembler which memory models to use. In 32-bit programs, we use the flat memory model which is associated with the processor's protected mode.

## \* .386

model flat.

• stack 4096

Exit processor proto

dword exit code: DWORD

The .386 directive identifies it as a 32-bit program. Line 2 uses the flat memory model, and window requires the model conversion to be used.

Line 3 sets aside 4096 bytes of storage.

Line 4 declares a proto type for the Exit process function.

## \* .CODE

• code

main proc

it is the beginning of the code section of the program (meaning)

(16)

what's after word is usually  
the main procedure

### \* .DATA

- The DATA directive creates a new data segment.
- This DATA segment contains the frequently used data for your program.
- DATA segment can occupy
  - up to 64k in MS-DOS
  - or up to 512 megabyte under file system model in Windows NT.

### \* ~~photo~~ PROTO

The proto directive by using  
the invoke directive

### Syntax

- Label ~~pt~~ PROTO [*'distance'*]  
[*'language-type parameter'*]: *tag*]