M-Khalid

23K-0727 Assignment 1 3J Date ____ 20__

Question 1:-

a) **High Level languages:-**

- High level languages are closer to human languages so they are easier to learn, read and write
- High level languages are machine independent so code written in them can run across different hardwares without much changes
- It makes use of English statements for operation

b) **Assembly language:-**

- It is a low level programming language just above machine code so it is difficult to learn
- It is a machine dependent language as it runs on specific processor architecture (e.g x86).
- It uses mnemonics to perform operations

Similarities:-

- Both HLL and Assembly need to be compiled or interpreted into machine code for computer to understand
- Both uses symbolic representations to refer to memory locations, data etc

Relationship:-

- High level languages has a one-to-many relationship with assembly language
- As a single High level language instruction expands into multiple assembly language instructions

b) Assemblers	Compilers
• Translates assembly code into machine code	• Translates High level language code into machine code or assembly code
• performs one-to-one translations	• Performs complex optimizations, error checking and code analysis
• Debugging is difficult in assembler	• Debugging is easier as compared to assembler

Date ____ 20__

()·High level languages are more portable because they are machine independent. The same code can run on different hardware for e.g A Python program written on windows can run on Linux as long as it has Python interpreter installed.
·Whereas Assembly language is machine dependent such as code written on x86 cannot run on ARM without much modification.
·High level languages are translated through compilers such as Java programs can run on any machine with Java virtual machine making it a highly portable language

Question 2:-

·Virtual machine is a software program that emulates the functions of some other physical or virtual computer
·The level 0 or VM(0) is the actual hardware or machine language of the micro-controller of arduino
·The level 1 or VM(1) is the assembly language used to program the microcontroller which is generated by compiling C code into assembly.
Compilation:- Compilation occurs at VM(1) which is level (2) assembly language
Translation:- Translation occurs at VM(0) which is the hardware level

Question 3:-

a) Last 4 digits of roll number = 0727
MOV EAX F0F7F2F7h
Add EAX 10000100h
Converting both hexadecimal values into binary:
F0F7F2F7 :- 1111 0000 1111 0111 1111 0010 1111 0111
10000100:- 0001 0000 0000 0000 0000 0001 0000 0000
Now adding both values:

Date _____ 20___

```
 1111 0000 1111 0111 1111 0010 1111 0111
+0001 0000 0000 0000 0000 0001 0000 0000
 0000 0000 1111 0111 1111 0011 1111 0111
```

Carry flag: 1

Overflow flag: 0

Zero flag: 0

Sign flag: 0

Parity flag: 0

Auxiliary carry flag: 0

b) x86 processor can access 1MB ram at a time because it uses 20 bit addresses ranging from 0 to FFFF to access the memory

c) (Segment x 10h) + offset

(12AB x 10h) + 025F

12AB0 + 025F

```
12AB0 in Binary:    0001 0010 1010 1011 0000
025F in Binary:   +           0000 0010 0101 1111
Add:                0001 0010 1100 1100 1111
```

Answer: 12CCF

d) • As the 8086 processor is a 16-bit architecture it uses 16 bit registers for efficiency and simplicity.

• A 16 bit segment value is placed in segment register the CPU automatically converts a 16bit segment value and a 16 bit offset value in 20 bit linear address

Question 4:-

```
INCLUDE Irvine32.inc
  .data
    Sunday = 0
    Monday = 1
    Tuesday = 2
    Wednesday = 3
```

Date ____ 20__

Thursday = 4
Friday = 5
Saturday = 6
DaysArray DWORD Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

```
.code
main PROC
call DumpRegs
call WriteDec

exit
main ENDP
END main
```

Question 5:-

```
INCLUDE Irvine32.inc
.data
Var1 DWORD 5 DUP (?)
Var2 BYTE 2 DUP (?)
Var3 BYTE 15 DUP ("&")
Var4 BYTE 7 DUP ("%")
Var5 BYTE 1 DUP ('M')

.code
main PROC
call DumpRegs
call WriteDec

exit
main ENDP
END main
```

Date _____ 20___

Question 6:-

i) 88h in binary : 1000 1000
90h in binary : +1001 0000
1 0001=1 1000=8

al = 18h, OF = 1, CF = 1 as the output generates a carry and the result is larger than the registers size

ii) 5 in binary : 0000 0101
123 in binary : 0111 1011
1000 0000

al = 123, SF = 1, OF = 1, CF = 0 since the MSB changes sign changes so signflag is 1 and also overflowflag is 1 because the result 128 is interpreted as a negative number in 8-bit signed arithmetic

Question 7:-

1) eax = dwList it would move 2 bytes from start due to DWORD label
eax = 20001000

2) ebx = [dwList+1] it would take an offset of 1 byte which are 00 from 3000 and store 2 bytes
ebx = 00200100

3) ecx = [dwList+2] it would take an offset of 2 bytes which is 3000 and store 2 bytes
ecx = 30002000

4) edx = [dwList+3] it would take an offset of 3 bytes
edx = 11300020

eax = 20001000h
ebx = 00200100h
ecx = 30002000h
edx = 11300020h