National University of Computer & Emerging Sciences, Karachi.
FAST School of Computing,
Mid Term II, Fall 2021.
November 24, 2021, 8:30 am – 9:30 am

Course Code: EE 2003	**Course Name:** Computer Organization and Assembly Language
Instructors: Dr. Nouman M Durrani, Shoaib Rauf, Aashir Mahboob, Aamir Ali, and Qurat ul Ain	
Student's Roll No:	**Section:**

Instructions:

- Except for your Roll No and Section, DO NOT SOLVE anything on this paper.
- Return the question paper.
- Read each question completely before answering it. There are **4 questions on 2 pages.**
- In case of any ambiguity, you may make an assumption. But your assumption should not contradict any statement in the question paper.
- All the answers must be solved according to the SEQUENCE given in the question paper, otherwise, points will be deducted.
- This paper is subjective.
- Where asked for values, only provide the **hex-decimal** values.
- Problems needing iterations should be coded using iterative instructions. No points will be awarded otherwise.

Time Allowed: 60 minutes. **Maximum Points**: 25 points

==

Q. No. 1 Briefly answer each of the following: [5 x 2 = 10 points]

(i) Differentiate between the arithmetic and logical shifts with one example instruction each.

Arithmetic shift is used for signed number multiplication and division

Logical shift is used for unsigned number multiplication and division.

(ii) Using shift and add instructions, multiply a number X_{10} by 43_{10}.

Let x be a number i-e, 5. Now we shift this number 5 to right 5 times, 3 times, 1 time and 0 time. After shifting each number we also add the shift of First number into the previous one

(iii) What will happen, if immediately upon entering a subroutine you execute a "POP" instruction?

The function lost its return address and will never the control to the main function

(iv) How the stack pointer is affected when a: (i) **Ret** and (ii) **Ret n** instruction is executed? Draw a stack frame to support your answer.

In Ret: the function directly returns to the address of main function next line where it is called

In Ret n: the function first clears all the local variables from the stack and then return to the main function next line where it is called.

(v) When does a divide overflow occur at the machine level? Give example instructions to illustrate.

```
mov ax,1000h
mov bl,10h
div bl ; AL cannot hold 100h
```

Q. No. 2 Consider a CNIC consists of 13 hexadecimal digits as shown below: [5 points]

xxxxx-yyyyyyy-z ; For Example: 12345-6789123-4

For reference, also consider the following data definition:

CNIC DWORD 67891234h, 00012345h

The last digit ‘z’ represents gender of an individual, the 7 ‘y’ digits indicate family number and the initial 5 ‘x’ digits refer to the information regarding the residence. Write an assembly language program to extract the family and gender information from the CNIC, and store it into the memory.

```
INCLUDE Irvine32.inc
.data
CNIC DWORD 67891234h, 00012345h
gender DWORD ?
family DWORD ?
.code
main PROC
     MOV EDX, CNIC
     AND EDX, 0Fh
     MOV gender, EDX
     MOV EDX, CNIC
     SHR EDX, 4
     MOV family, EDX
     ;FOR CHECKING UNCOMMENT
     ;MOV ESI, OFFSET CNIC
     ;MOV EBX,4
     ;MOV ECX,4
     ;call dumpmem
exit
main ENDP
END main
```

Q. No. 3 Write an assembly code for a procedure named geometric sequence, that takes three arguments a, r, and b stored on the stack, and display the sequence on the console window: [5 points]

$$a, a * r, a * r^2, a * r^3, a * r^4, \ldots\ldots\ldots\ldots\ldots, b$$

$$3, 6, 12, 24, 48, 96, 192, 384,$$

Here a =3 is the starting number of a sequence, b = 750 is the upper bound (the sequence does not cross the limit and may or not be the part of the sequence) and r = 2 is the ratio between two consecutive terms. The procedure should be flexible for all kinds of parameters. Here, you are also supposed to draw the stack frame.

```
INCLUDE Irvine32.inc
.data
PRPA BYTE "Enter the value of a : ",0
PRPB BYTE "Enter the value of b : ",0
PRPR BYTE "Enter the value of r : ",0
```

```
PRPS BYTE ", ",0

.code
main PROC
     MOV EDX, OFFSET PRPA
     CALL WRITESTRING
     CALL READDEC
     PUSH EAX
     MOV EDX, OFFSET PRPR
     CALL WRITESTRING
     CALL READDEC
     PUSH EAX
     MOV EDX, OFFSET PRPB
     CALL WRITESTRING
     CALL READDEC
     PUSH EAX
     CALL GEOMETRIC_SEQUENCE

exit
main ENDP
GEOMETRIC_SEQUENCE PROC
     ENTER 0,0
     MOV EAX, [EBP+16]
     MOV ECX, [EBP+12]
     MOV EBX, [EBP+8]

     L1:
          CMP EAX, EBX
          JA L2
          CALL WRITEDEC
          MOV EDX, OFFSET PRPS
          CALL WRITESTRING
          MUL ECX
          JMP L1
     L2:

     LEAVE
     RET 12
GEOMETRIC_SEQUENCE ENDP
END main
```

Q. No. 4 Write an equivalent assembly language code for the following High Level Language code: [5 points]

```
int main()
{
     char array[] = "Assembly language is fun";
     int arraySize = sizeof(array)/ sizeof(array[0]);
     char iSearch = '\0';
     int  searchCount=0;

     cout<<"Enter a char = ";
     cin>>iSearh;

     int i = 0;
     while(arraySize>0)
    {
      if (iSearch == array[i])
           { searchCount++; }
      arraySize --;
    }
     cout<<"Char count = "<<searchCount;
     i++;
}
```

```
INCLUDE Irvine32.inc
.data
PRMP BYTE "Enter a char = ",0
PRMP2 BYTE "Char count = ",0
array BYTE "Assembly language is fun",0
arraySize DWORD LENGTHOF array
iSearch BYTE 0
searchCount DWORD 0
i DWORD 0
.code
main PROC
     MOV EDX, OFFSET PRMP
     CALL WRITESTRING
     CALL READCHAR
     CALL WRITECHAR
     CALL CRLF
     MOV iSearch, AL
     L1:
          CMP arraySize, 0
          JBE L2
          MOV EBX, i
          CMP array[EBX], AL
          JNE L3
                INC searchCount
          L3:
          DEC arraySize
          INC i
          JMP L1
     L2:
     MOV EDX, OFFSET PRMP2
     CALL WRITESTRING
     MOV EAX, searchCount
     CALL WRITEDEC
exit
main ENDP
END main
```

Best of Luck