MPI Lab Question Bank

1. ALP to convert from ºC to ºF

- Store the value in ºC which has to be converted into *F in the data segment (part of memory).
- Use a variable name for this location as Deg_Cent in the assembly program.
- Create an empty space in the data segment to store the *F value. Use variable name for this location as Deg_Far in the assembly program
- Access the data segment to get the value Deg_Cent
- Implement the formula for *C to *F with the help of arithmetic instructions
- Store the final F in the data segment in a location mapped to Deg_Far

2. ALP to convert from ºF to K

- Store the value in ºF which has to be converted into K in the data segment (part of memory).
- Use a variable name for this location as Deg_Far in the assembly program.
- Create an empty space in the data segment to store the K value. Use variable name for this location as Deg_Kel in the assembly program
- Access the data segment to get the value Deg_Far
- Implement the formula for *F to K with the help of arithmetic instructions
- Store the final K in the data segment in a location mapped to Deg_Kel

3. ALP to find factorial of a number

- Prompt the user to enter a number between 1 and 8 (you can show a message "Enter the number between 1 and 8")
- Once the user enters the number you can compute the factorial of the number and store it in data segment (part of memory) which is mapped to a variable 'FACT'

3. ALP to find factorial of a number using recursion

- Prompt the user to enter a number between 1 and 8 (you can show a message "Enter the number between 1 and 8")
- Once the user enters the number you can compute the factorial of the number using recursion and store it in data segment (part of memory) which is mapped to a variable 'FACT'

4. ALP to find sum of first n natural numbers

- Prompt the user to enter a number between 1 and 20 (you can show a message "Enter the number between 1 and 20")
- Once the user enters the number you can compute the sum of the first n natural numbers and store it in data segment which is mapped to a variable 'NSUM'

5. ALP to find the first n terms of the fibonacci series

- Store a byte value in the memory location defined by variable “N”
- Create an empty array of N bytes in the memory, with array name as “FIB_Out”
- Generate N terms of Fibonacci series and store them in array location “FIB_Out”

6. ALP to sort the elements of an array in ascending order using bubble sort

- Store a byte value representing the number of elements in an array at a memory location defined by variable “N”
- Store array of “N” bytes in the memory, with array name as “ARRAY”
- Sort the elements of array “ARRAY” in ascending order using Bubble sort algorithms

7. ALP to sort the elements of an array in descending order using bubble sort

- Store a byte value representing the number of elements in an array at a memory location defined by variable “N”
- Store array of “N” bytes in the memory, with array name as “ARRAY”
- Sort the elements of array “ARRAY” in descending order using Bubble sort algorithms

8. ALP to find the GCD of two numbers

- Store a byte value in the memory location defined by variable “Num1”
- Store a byte value in the memory location defined by variable “Num2”
- Create an empty memory location defined by variable “GCD_Out”
- Find GCD of two numbers in “Num1” and “Num2” and store the result in the memory location pointed by GCD_Out

9. ALP to find the LCM of two numbers

- Store a byte value in the memory location defined by variable “Num1”
- Store a byte value in the memory location defined by variable “Num2”
- Create an empty memory location defined by variable “LCM_Out”
- Find LCM of two numbers in “Num1” and “Num2” and store the result in the memory location pointed by LCM_Out

10. ALP to print a rectangular pattern using *

- Prompt the user to enter a number between 1 and 9 (you can show a message “Enter the number between 1 and 9”)
- Once the number is entered, print a rectangular pattern with “*” character, such that for given N, value N rows are printed with each row containing N “*”. For example, if the number entered is 3 then the following pattern should be printed,

```
***
***
***
```

11. ALP to print a right triangular pattern using *

- Write ALP to accept a input value between 1 to 9 from user and print a right triangular “*” pattern.
- For example, if the number entered is 3 then the following pattern should be printed,

```
*
**
***
```

12. ALP to print a spaced triangular pattern using *

- Write ALP to accept a input value between 1 to 9 from user and print a triangular “*” pattern.
- For example, if the number entered is 3 then the following pattern should be printed,

```
  *
 * *
* * *
```

13. ALP to copy a character string from one location to another and display both

- Store the string length in the memory location defined by variable name “StrLen”
- Store a character string in the memory location defined by variable “STR”
- Reserve memory defined by variable “STRCPY” for storing the copied string
- Copy the string from location “STR” to “STRCPY” and display both the strings on the console in two different lines.

14. ALP to reverse a string, and print both the original and new versions of the string

- Store the string length in the memory location defined by variable name “StrLen”
- Store a character string in the memory location defined by variable “STR”
- Reserve memory defined by variable “REVSTR” for storing the reversed string
- Reverse the string at location “STR” and store it at “REVSTR”. Display both the strings on the console in two different lines.

15. ALP to check if a given string is a palindrome

- Store the string length in the memory location defined by variable name “StrLen”
- Store a character string in the memory location defined by variable “STR”
- Check if the given string is a palindrome, and if it is, display a message accordingly in the console, showing the original and reversed strings to be equal. If it is not the same, display a message indicating the same and display both original and reversed strings to indicate the difference.

16. ALP to reverse a character string without using any additional memory locations, and display both original and reversed strings

- Store the string length in the memory location defined by variable name “StrLen”
- Store a character string in the memory location defined by variable “STR”
- Display the original string.

- Reverse the string at location “STR” and store it at the same location.
- Display the reversed string.

17. ALP to print a dense triangular pattern using character *

- Prompt the user to enter a number between 1 and 9 (you can show a message “Enter the number between 1 and 9”)
- Once the number is entered, print a triangular pattern with “*” character. If the number entered is 3 then the following pattern should be printed. (first row will have 2 spaces followed by one *, second row will have 1 space followed by 3 “*” and last row will have no spaces followed by 5 “*”)

```
  *
 ***
*****
```

19.Write ALP to check if the string (of 20 bytes) stored at location 5000h:0000h is the same as the string stored at 6000h:0000h. Store 1 in BL if strings are equal else store 0 in BL.

20.Find multiplication of two 8 bit numbers without using MUL operator.

- Hint: use shift operators.

21.Find the roots of a quadratic equation given Δ is a perfect square.

- Create a function to find the square root of the perfect square.(Repeated subtraction method)
- Call the above mentioned function in the main module.

22.Write an ALP code to find the square root of a perfect square.(Repeated Subtraction method).

23.Write a program in 8086 ALP to add data in one block of memory with data in another block of memory (Array Addition)
Assume,

No of elements in the block: 5
Block 1: 23h, 42h, 63h, 77h, 25h
Block 2: 31h, 12h, 50h, 33h, 20h
After the program Block 2 should be as below:
Block 2: 54h, 54h, B3h, AAh, 45h

24.Write an assembly language program to convert 8-bit binary data to BCD. Assume the input data is stored in the data segment at offset address 1100h.

25.An ALP program shown below calls a procedure BCDtoBin, which converts BCD number stored in AL into binary number. The procedure stores the result in AL. Write the procedure for BCDtoBin in ALP

```
 .MODEL SMALL
 .DATA
BCDNum db 45h
 .CODE
 .STARTUP
MOV AL, BCDNu
CALL BCDtoBin
 .EXIT
 .END
```

26.Write an assembly language program to convert 8-bit binary data to Gray Code. Assume the input data is stored in the data segment at offset address 1100h.