**Assignment No. 4**

# AIM:-

Write menu driven ALP to perform string manipulations. The strings to be accepted from the user isto be stored in code segment Module\_1 and write FAR PROCEDURES in code segment Module\_2 toperform any two of the following string operations:

i. Concatenation of two strings.

ii. Comparison of two strings.

# OBJECTIVES:-

1. To study use of FAR procedures.
2. To study linking of 2 or more modules into one executable module.

**THEORY :-**

**ASSEMBLER DIRECTIVES USED :-**

1.PUBLIC – This directive indicates that the procedure or variable declared public is defined in the same segment but may be used in some other segment at link time.

Format – PUBLIC name of procedures or variables

2.EXTRN – This directive indicates that the procedure or variable marked external is not defined in this segment/program i. e. it is external but will be used in this program.

Format – EXTRN name of the procedure:far

Format – EXTRN type(Byte/word/..) name of the variable

**LIST OF PROCEDURES USED :-**

1. Extern Concatenation → It is used to concatenate two strings.

Input : 2 strings.

Output : concatenated string.

1. Extern compare → It is used to compare 2 strings.

Input : 2 strings.

Output : It states whether the given strings are equal or not.

1. Extern Substring → It is used to determine whether the string is a substring of the first string or not.
2. Extern Text → It is used to accept paragraph & find no. of capital letters, words, lines.

**ALGORITHM FOR MAIN PROGRAM :**

1. Physical initialization of data segment.
2. Using Macro display menu. 1. Accept 2. Concatenate 3. Compare 4. Substring 5. Text 6. Exit
3. Accept choice from the user.
4. Is choice = 1, accept the string. Set flag = 01.
5. Is choice = 2 jump to step 11 else go to step 6.
6. Is choice = 3 jump to step 12 else go to step 7.
7. Is choice = 4, jump to step 13 else goto step 8.
8. Is choice = 5, jump to step 14 else goto step 9.
9. Is choice = 6, jump to step 15 else goto step 10.
10. Display wrong choice.
11. If flag = 01, call procedure Concatenate else goto step 2.
12. If flag = 01, call procedure compare else goto step 2.
13. If flag = 01, call procedure substring else goto step 2.
14. If flag = 01, call procedure text else goto step 2.
15. Terminate the program and stop.

**ALGORITHM FOR CONCATENATION :**

1. Initialise one memory pointer(SI) to the next of last location of the first string.
2. Initialize another pointer (DI) to the first location of second string.
3. Copy the contents pointed by DI to the register.
4. Copy the contents of that register to the location pointed by SI & update both pointers.
5. Repeat step 3 & 4 until the second string gets copied.
6. Display the resultant string.
7. Return.

**ALGORITHM FOR COMPARE PROCEDURE:**

1. Initialize two pointers to first location of both the strings.
2. Compare the ASCII values of contents pointed by SI & DI.
3. If equal, then increment both the pointers.
4. If not equal, display “ NOT EQUAL” message & return.
5. If Step 3 is executed decrement the length counter of one of strings & jump to step 2 if not zero.
6. If comparison is finished, display “ Strings are equal”.
7. Return.

**CONCLUSION:**