# LABORATORY 5

TITLE OF THE LABORATORY EXERCISE: SEARCHING AN ELEMENT IN AN ARRAY

1. INTRODUCTION AND PURPOSE OF EXPERIMENT

   STUDENTS WILL BE ABLE TO PERFORM SEARCH OPERATIONS IN AN ARRAY OF INTEGERS OR CHARACTERS

2. AIM AND OBJECTIVES

   AIM

   TO DEVELOP ASSEMBLY LANGUAGE PROGRAM TO PERFORM SEARCH OPERATIONS IN AN ARRAY

   OBJECTIVES

   AT THE END OF THIS LAB, THE STUDENT WILL BE ABLE TO

   - IDENTIFY INSTRUCTIONS TO BE USED IN ASSEMBLY LANGUAGE
   - PERFORM SEARCH OPERATIONS IN ASSEMBLY LANGUAGE

3. EXPERIMENTAL PROCEDURE

   1. WRITE ALGORITHM TO SOLVE THE GIVEN PROBLEM
   2. TRANSLATE THE ALGORITHM TO ASSEMBLY LANGUAGE CODE
   3. RUN THE ASSEMBLY CODE IN GNU ASSEMBLER
   4. CREATE A LABORATORY REPORT DOCUMENTING THE WORK

4. QUESTIONS

   DEVELOP AN ASSEMBLY LANGUAGE PROGRAM TO PERFORM THE FOLLOWING:

   1. SEARCHING AN ELEMENT IN AN ARRAY OF 'N' NUMBERS
   2. READ A SENTENCE WITH AT LEAST ONE SPECIAL CHARACTER AND SEARCH FOR THE SPECIAL CHARACTER AND PRINT IT. E.G., CONSIDER THE INPUT {YOUREMAILID@MSRUAS.AC.IN }OUTPUT: @, .
   3. DEVELOP AN ASSEMBLY LANGUAGE PROGRAM TO COMPUTE THE PARITY OF A HEXADECIMAL NUMBER STORED IN THE REGISTER1. IF REGISTER1 HAS ODD

NUMBER OF ONES, UPDATE REGISTER2 WITH 0X01. IF REGISTER1 HAS EVEN NUMBER OF ONES, UPDATE REGISTER2 WITH 0X00.

NOTE: REGISTER1 AND REGISTER2 CAN BE ANY GENERAL PURPOSE REGISTERS.

5. CALCULATIONS/COMPUTATIONS/ALGORITHMS


FIG 1 PROGRAM TO SEARCHING AN ELEMENT IN AN ARRAY OF 'N' NUMBERS

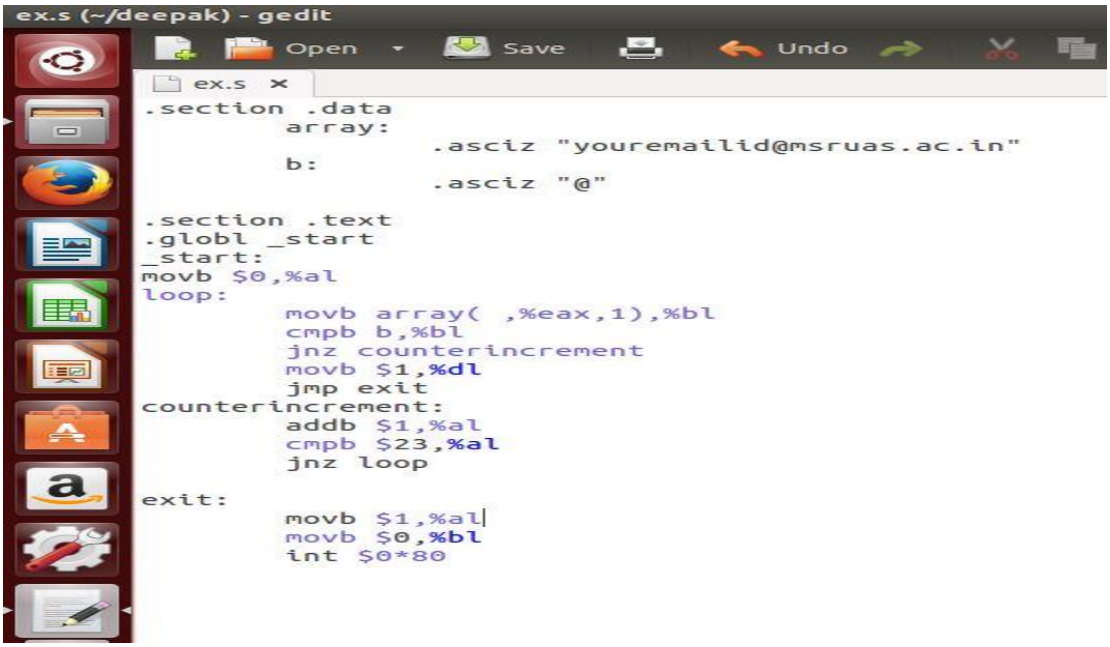
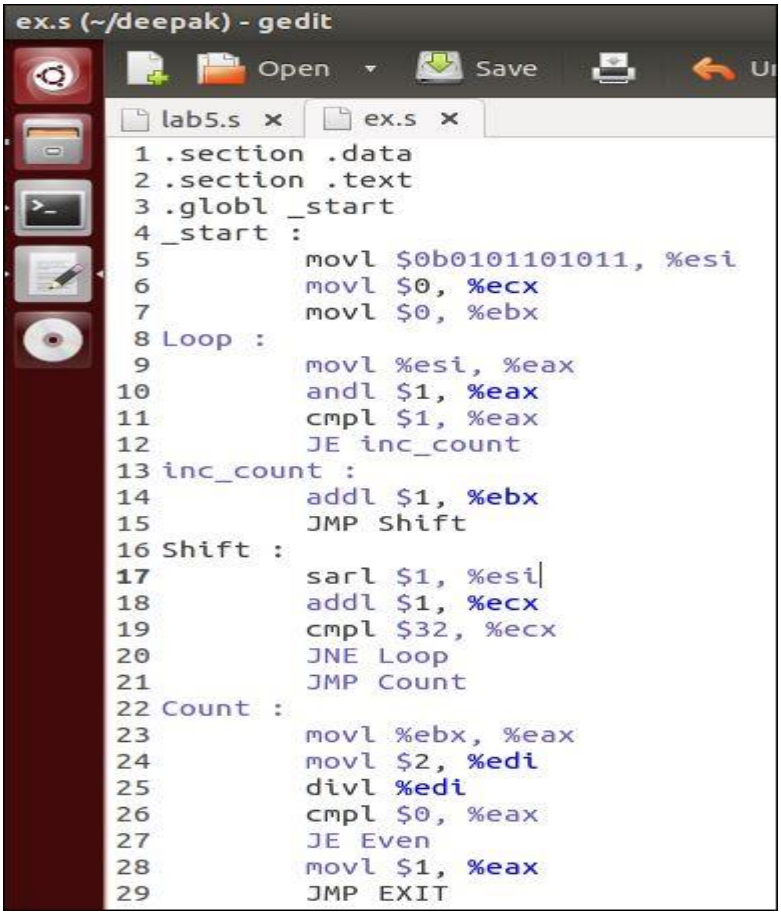



FIG 2 PROGRAM TO READ A SENTENCE WITH AT LEAST ONE SPECIAL CHARACTER AND SEARCH FOR THE SPECIAL CHARACTER AND PRINT IT.



```
30 Even :
31          movl $0, %eax
32 EXIT :
33          movl $1, %eax
34          movl $0, %ebx
35          int $0x80
```

FIG 3 PROGRAM AN ASSEMBLY LANGUAGE PROGRAM TO COMPUTE THE PARITY OF A BINARY NUMBER

6. PRESENTATION OF RESULTS

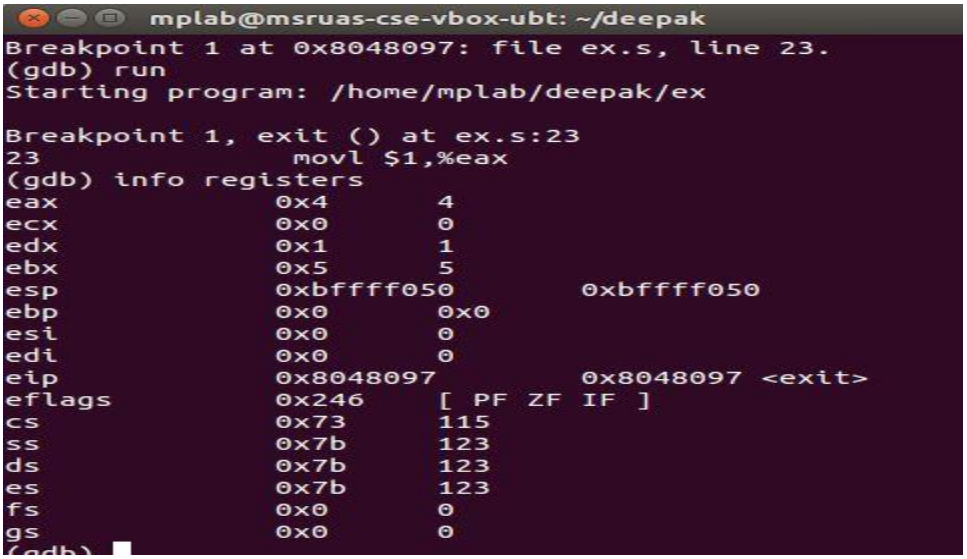



FIG 4 RESULT OF PROGRAM TO SEARCHING AN ELEMENT IN AN ARRAY OF 'N' NUMBERS

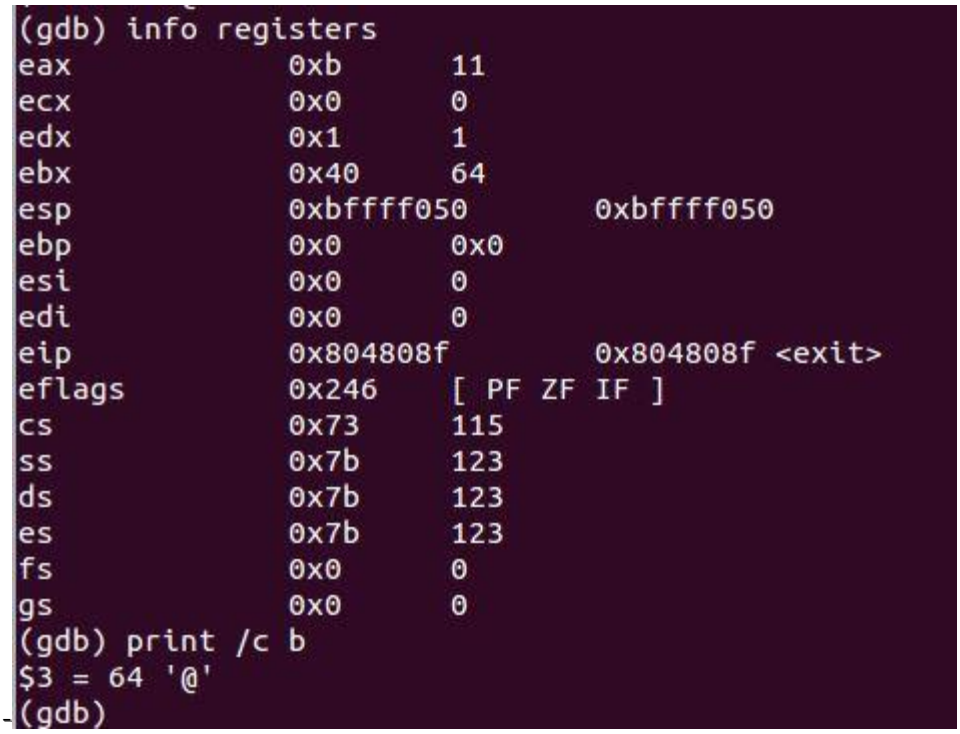



FIG 5 RESULT OF PROGRAM TO READ A SENTENCE WITH AT LEAST ONE SPECIAL CHARACTER AND SEARCH FOR THE SPECIAL CHARACTER AND PRINT IT.

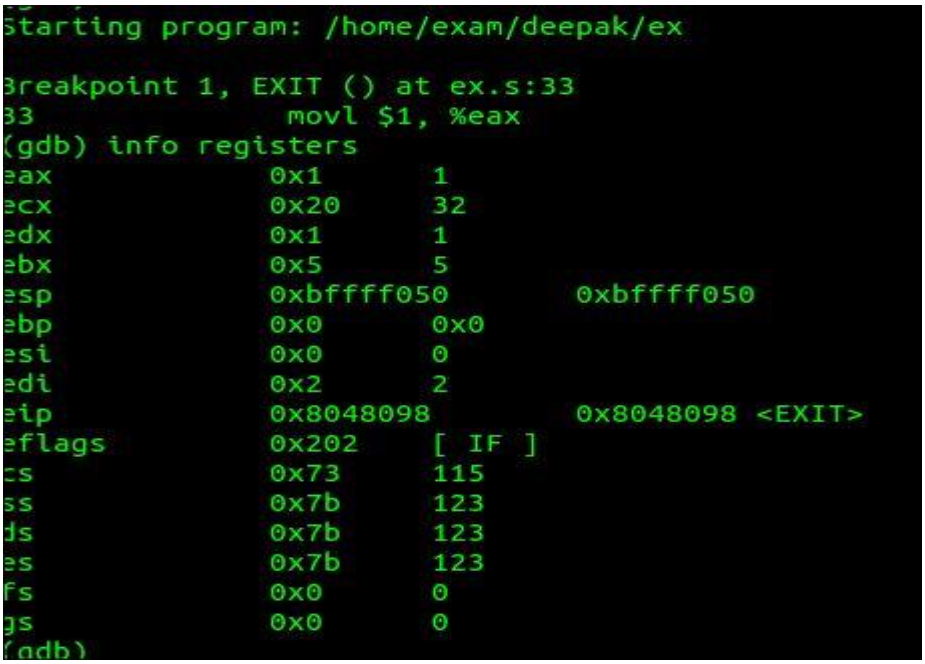



FIG 6 RESULT OF PROGRAM AN ASSEMBLY LANGUAGE PROGRAM TO COMPUTE THE PARITY OF A BINARY NUMBER

1. CONCLUSIONS

EXECUTION FLOW CAN BE CONTROLLED BY USING CONDITIONAL INSTRUCTIONS, WHICH INCLUDES A CMP INSTRUCTION FOLLOWED BY A JUMP INSTRUCTION, A CMP INSTRUCTION COMPARES THE TWO OPERANDS AND UPDATES THE FLAG REGISTER, THIS IS THEN USED WITH JUMP INSTRUCTION TO GO TO SOME OTHER PART OF THE PROGRAM, USING THIS WE CAN FORM LOOPING STRUCTURES TO DO STUFF LIKE SEARCHING ELEMENT IN ARRAY, SEARCHING SPECIAL CHACTER IN ARRAY, AND PROGRAM AN ASSEMBLY LANGUAGE PROGRAM TO COMPUTE THE PARITY OF A BINARY NUMBER

COMMENTS

1. LIMITATIONS OF EXPERIMENTS

ALTHOUGH LOOPING STRUCTURES CAN BE FORMED USING THE CMP, JCC INSTRUCTIONS BUT RECURSIVE STRUCTURES ARE COMPLEX TO FORM USING JUST THESE INSTRUCTIONS.

2. LIMITATIONS OF RESULTS

NONE

3. LEARNING HAPPENED

WE LEARNT THE USE OF COMPARE, UNCONDITIONAL JUMP AND CONDITIONAL JUMP INSTRUCTIONS TO FORM LOOPING STRUCTURES AND CONDITIONAL STATEMENTS.

4. RECOMMENDATIONS

SINCE A PROGRAM CAN CONTAIN NUMEROUS LOOP LABELS, EACH LABEL SHOULD BE CAREFULLY NAMES, AND THE PROGRAMMER MUST KEEP TRACK OF WHICH PARTS OF THE PROGRAM JUMP TO WHERE, ELSE THERE MIGHT BE CHANCES OF FORMING INFINITE LOOPS.

SIGNATURE AND DATE