**ASSIGNMENT 10**

**AIM**:-Write an ALP to compare two strings using string instructions/without using string instructions.

**THEORY**:-

String manipulation instructions:-

String is a contiguous block of bytes or words and can be used to hold any type of data or information that will fit into two bytes or words. Number of operations performed with strings. The 8086 microprocessor support strings instructions for string movement scan, comparison, load and Store.

Compare instruction:-

1. CMPS compare string

2. CMPSB compare string byte

3. CMPSW compare string bite

1. The instruction cmps compares a bite or a word in the source string with a byte or a word in the destination string.

2. The source must be in the data segment and destination must be in extra segment.

3. The offset of the source byte or a word must be placed in SI register, which is represented as DS:SI and offset of the destination bite or a word must be in D I register which is represented as ES:DI.

4. On the execution of this instruction Si and Di registers are automatically incremented by 1 to point next element of source and destination.

5. If the direction flag is reset the register SI and DI will be incremented by 1 byte compared and incremented by 2 for word compare.

6. If the direction flag is set DF=1, the register SI and DI I will be degree mented by one for bite compare and decremented by to forward compare.

7. The DS:SI and ES:DI register must be loaded prior to the execution ofCMPS of instruction.

9. Another way to compare a byte forward string is by using implicit instruction CMPSB and CMPSW.

10. The multiple bite or Word compare the count must be loaded in CX register which functions as a counter.

Flags modified:-AE,CF,DF,PF,SF,ZF.

Operations performed by the instructions:-

1. If destination string byte/word>source string byte/word then cf=0,zf=0,sf=1.

2. If destination string byte/word source string byte/wo then CF=1,zf=0,sf=1

3. Who is destination string byte / word = source string byte / word then

cf=0 and,zf=1,sf=0.

For byte comparison:-

1. If DF =0 THEN SI=SI+1,DI=DI+1

2. if DF=1 if thenSI n =SI-1,DI=DI-1

For word comparison

1. IF DF f=0 then SI=SI+2,DI=DI+2

2. DF=1 THEN SI=SI-2,DI=DI-2

Comparison of two strings without using string instructions:-

Algorithm:-

1. Initialise data segment.

2. Find the length of destination string.

3. Compare length of both the strings.

5. If length of both strings are not same then go to step 8.

6. Display message 'strings are same'

7. Stop

8.' display message strings are not same'

9. Stop. Comparison of two strings using string instructions:-

Algorithm:-

1. Initialise data and extra segment that is D S and ES

2. Find the length of source string.

3. Find the length of destination string.

3. Compare length of both the strings.

4. Compare length of both the strings.

5. If length of both strings that are not same then go to step 10.

6. Compare strings character by character.

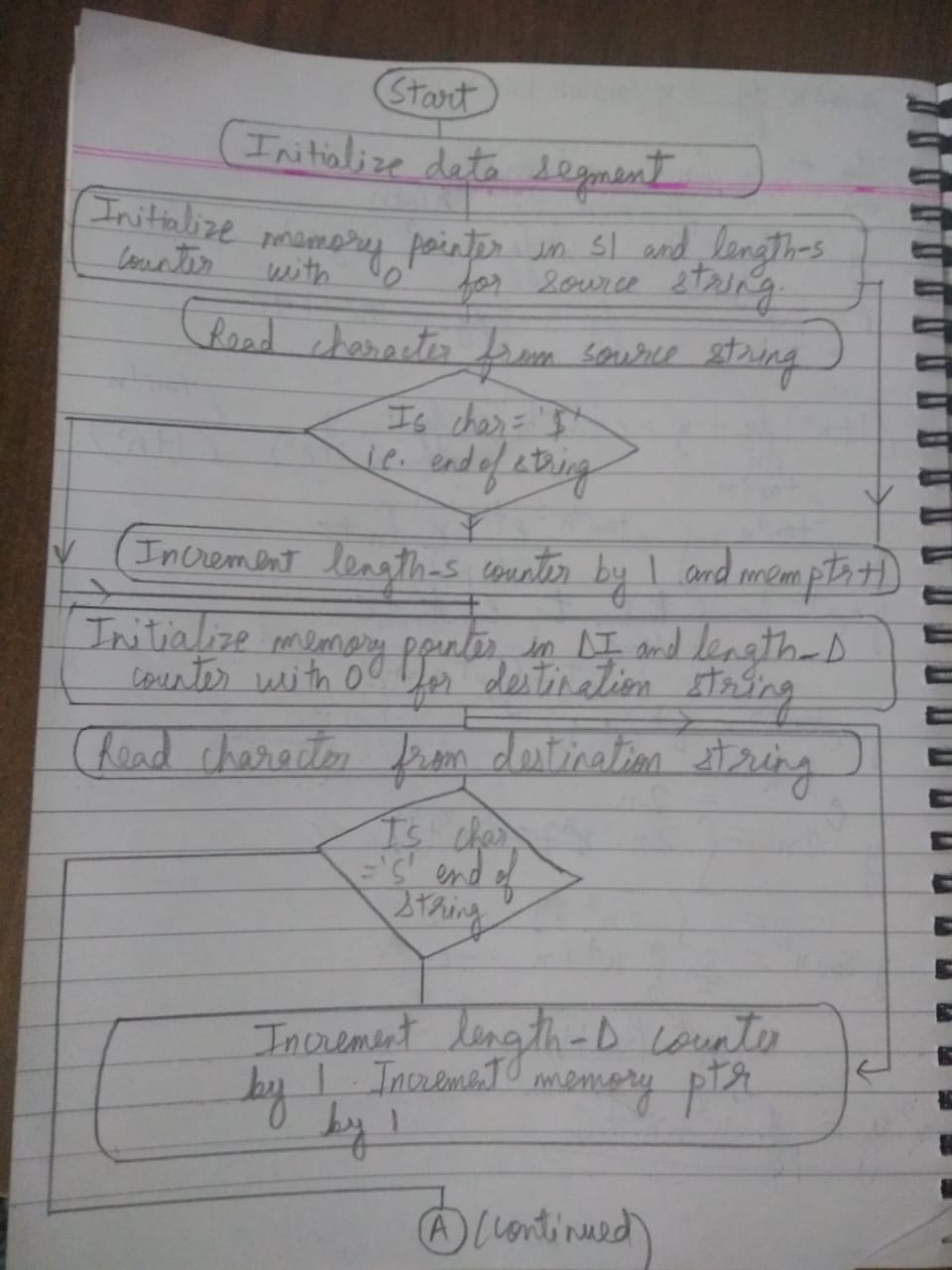
7. If characters of both the strings are not same then go to step 10.

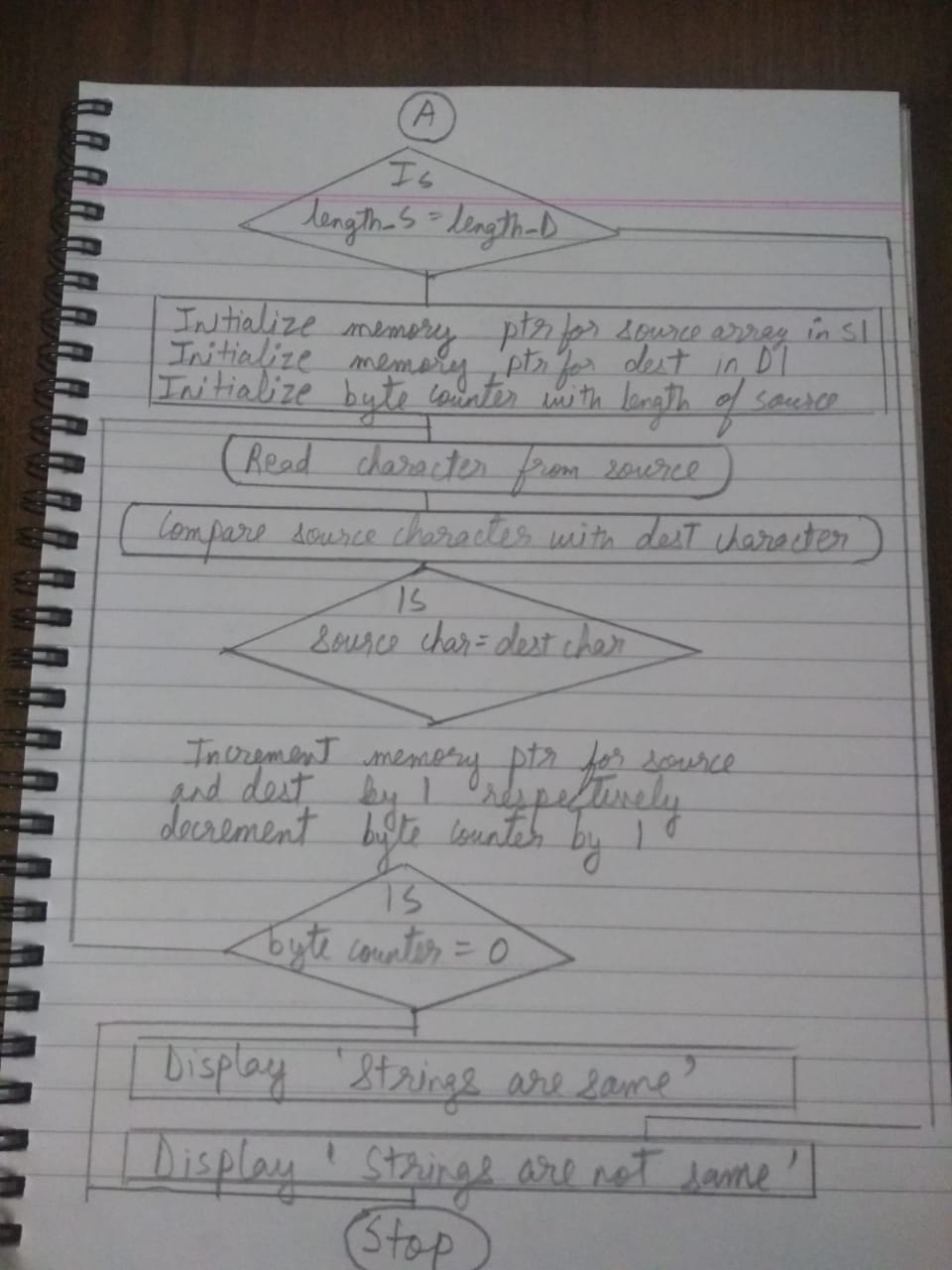
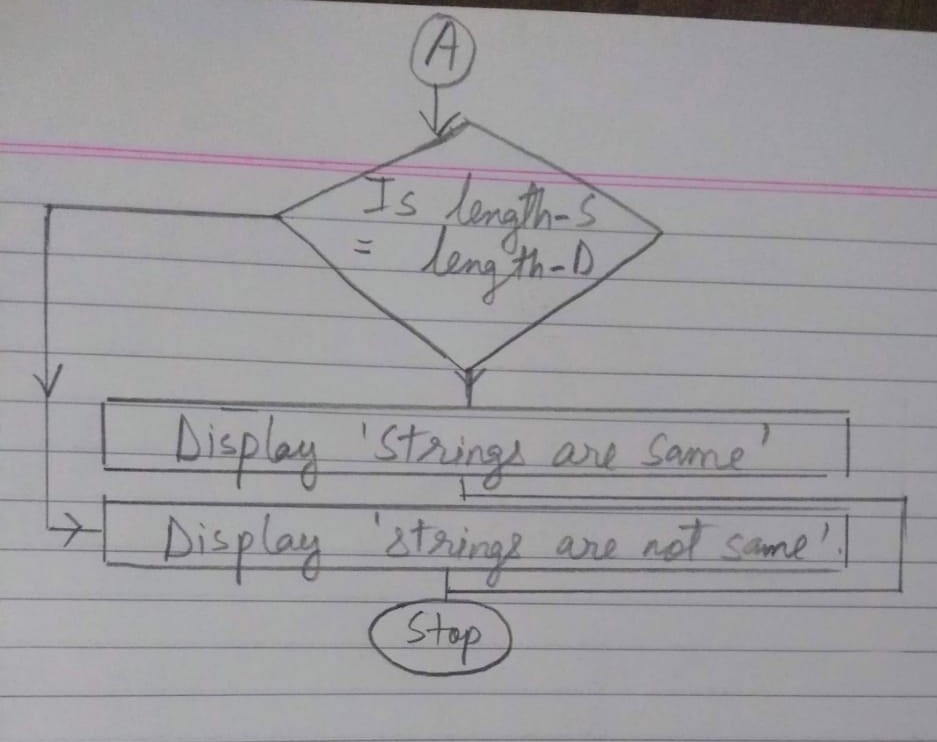
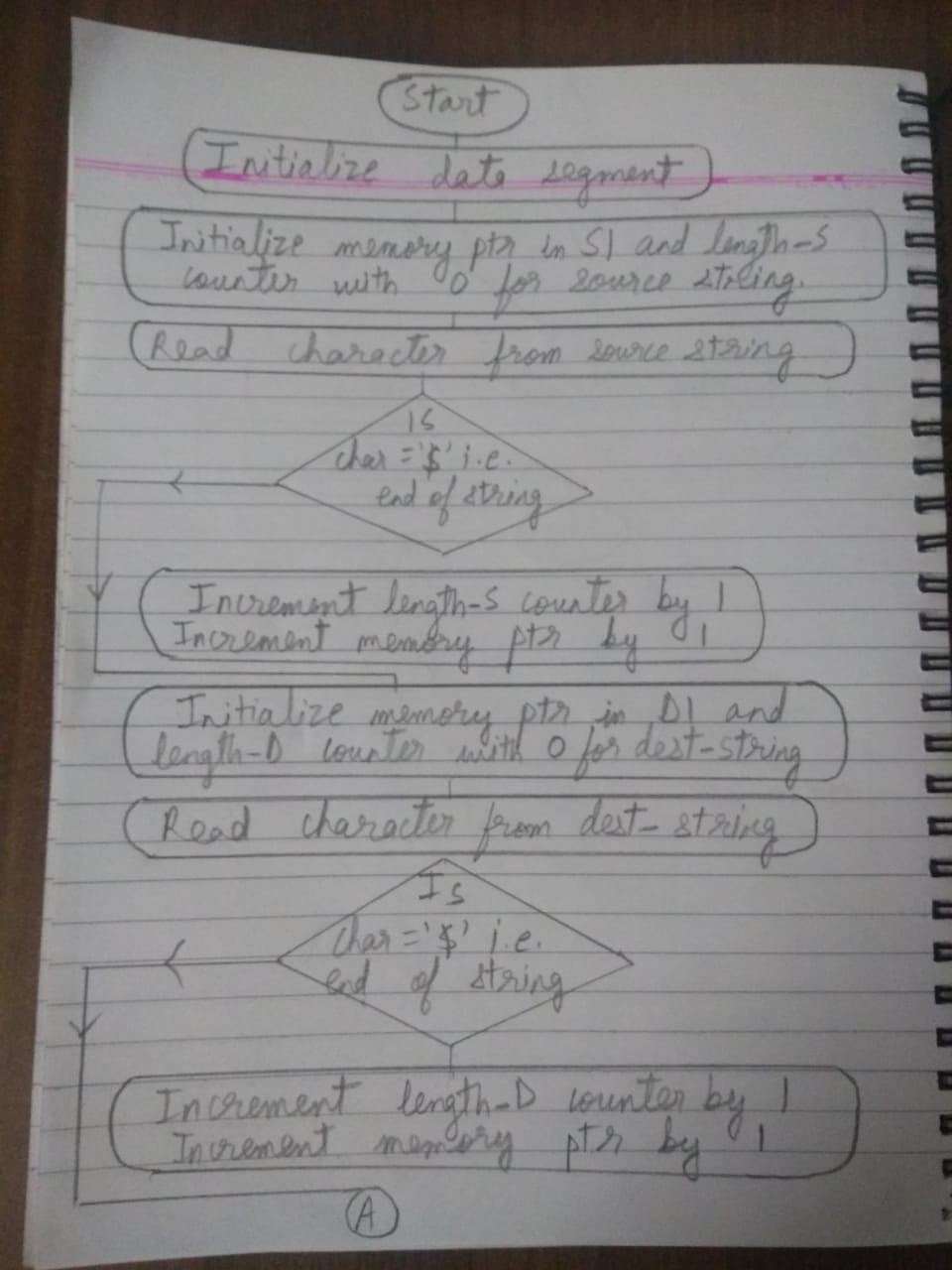
8. Display message 'strings are same'.

9. Stop

10. Display message' strings are not same'.

11. Stop.





**CONCLUSION:** Hence, we have written an ALP to compare two strings using string instructions/without using string instructions.

**ASSIGNMENT 11**

**Aim**:-

Write an ALP to display string in reverse order, string length, concatenation of two strings.

**THEORY**:-

Microprocessor: A microprocessor is a component that performs tasks and instructions involved in computer processing. In a computer system the MP is the central unit that executes and manages the logical instructions passed to it.

MP may also be called a processor or CPU that it is actually more advanced in terms of architectural design and is built across Silicon microchip.

Algorithms and flowcharts:

A. Display string in reverse order

1. Initialise data segment.

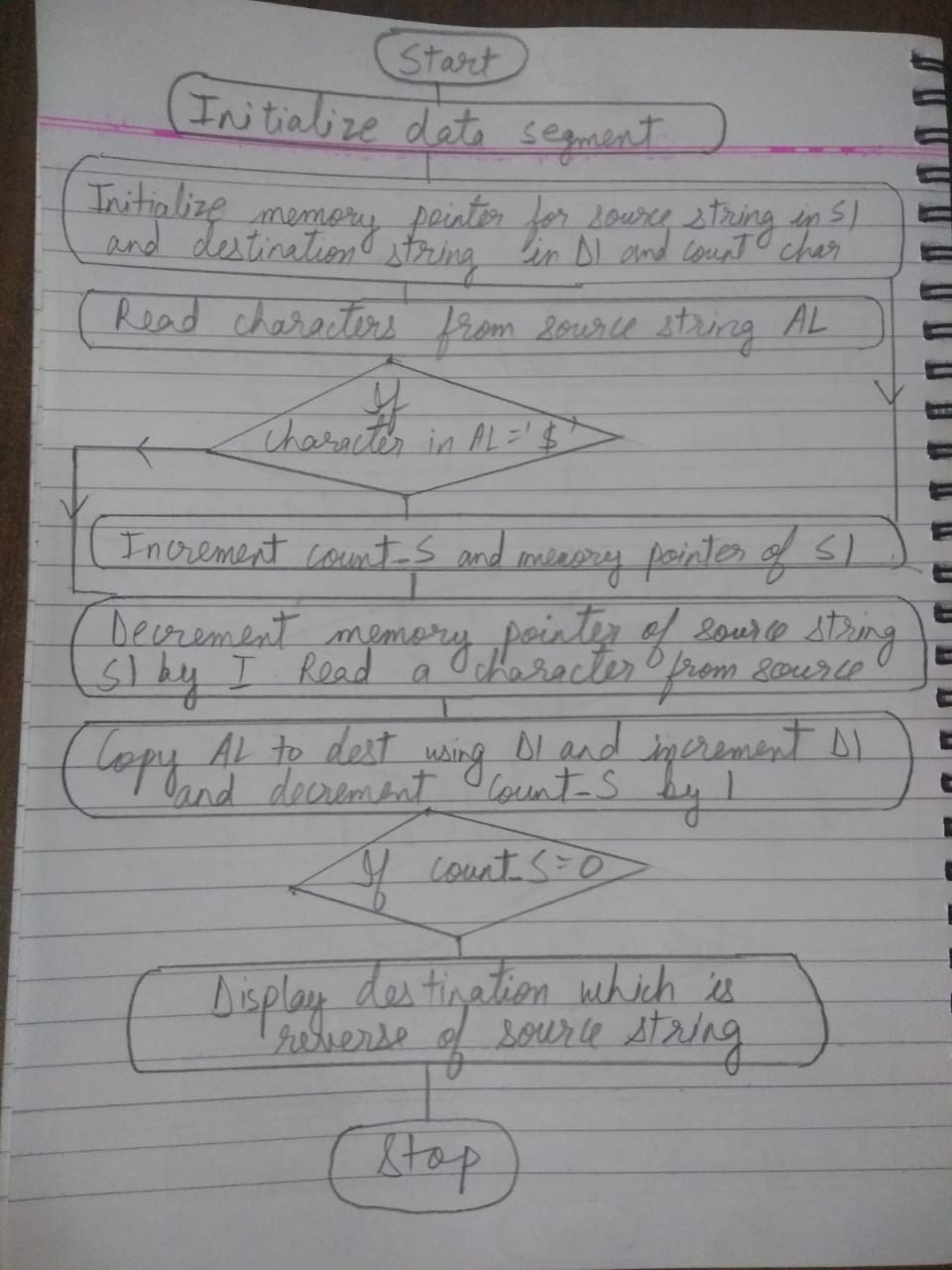
2. Find length of source string.

3. Copy source string to destination string in Reverse.

4. Display source and destination string.

5. Stop.

Flowchart:-



B. Algorithm to find length of string:-

1. Initialise data segment.

2. Initialise length counter, memory counter to read characters.

3. Read characters from string.

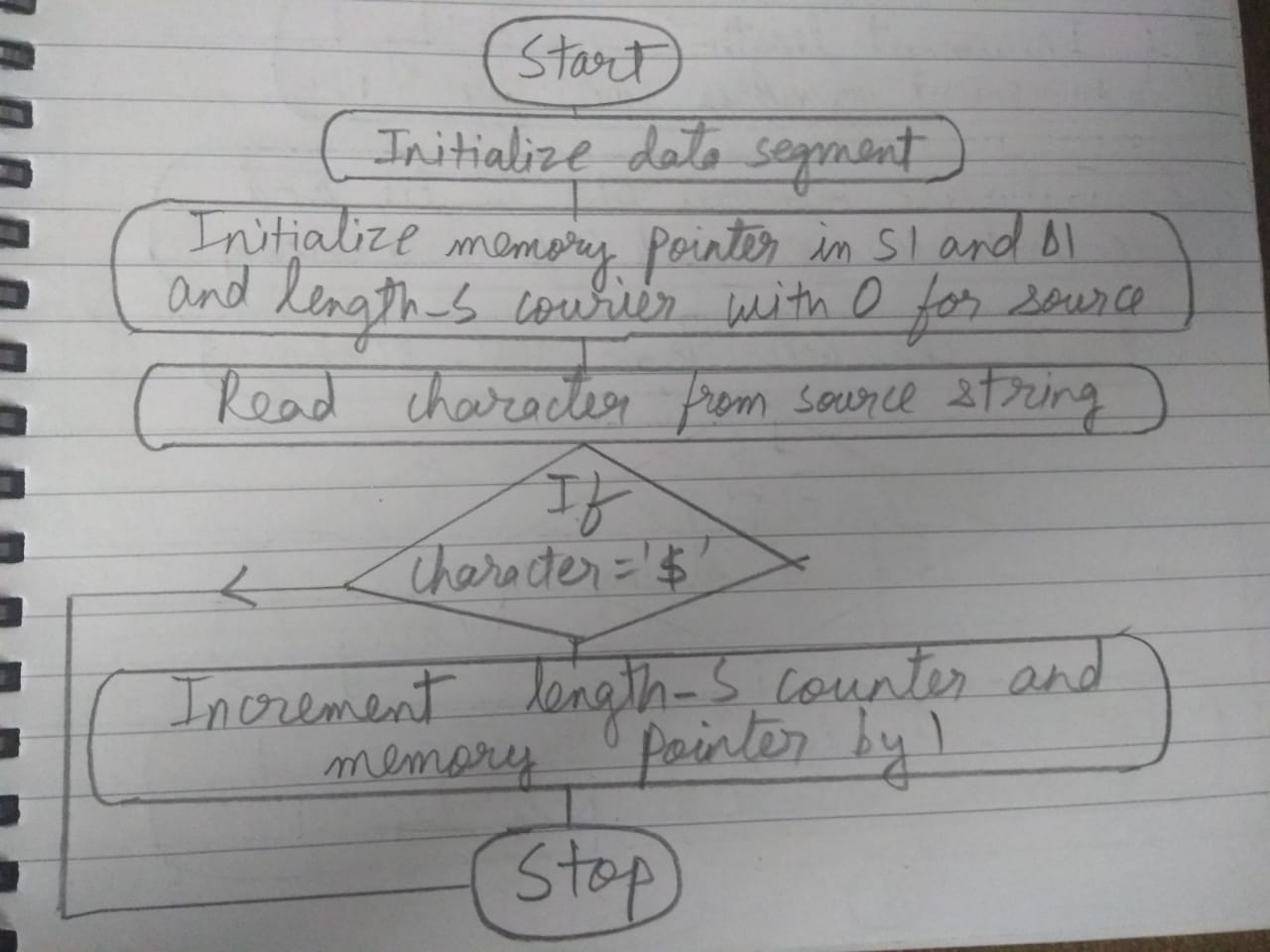
4. If is'$' then go to step 7.

5. Increment length counter and memory to next character.

6. Go to step 3.

7. Stop.

Flowchart:-



C. Concatenation of two strings algorithm:-

1. Initialise data segment.

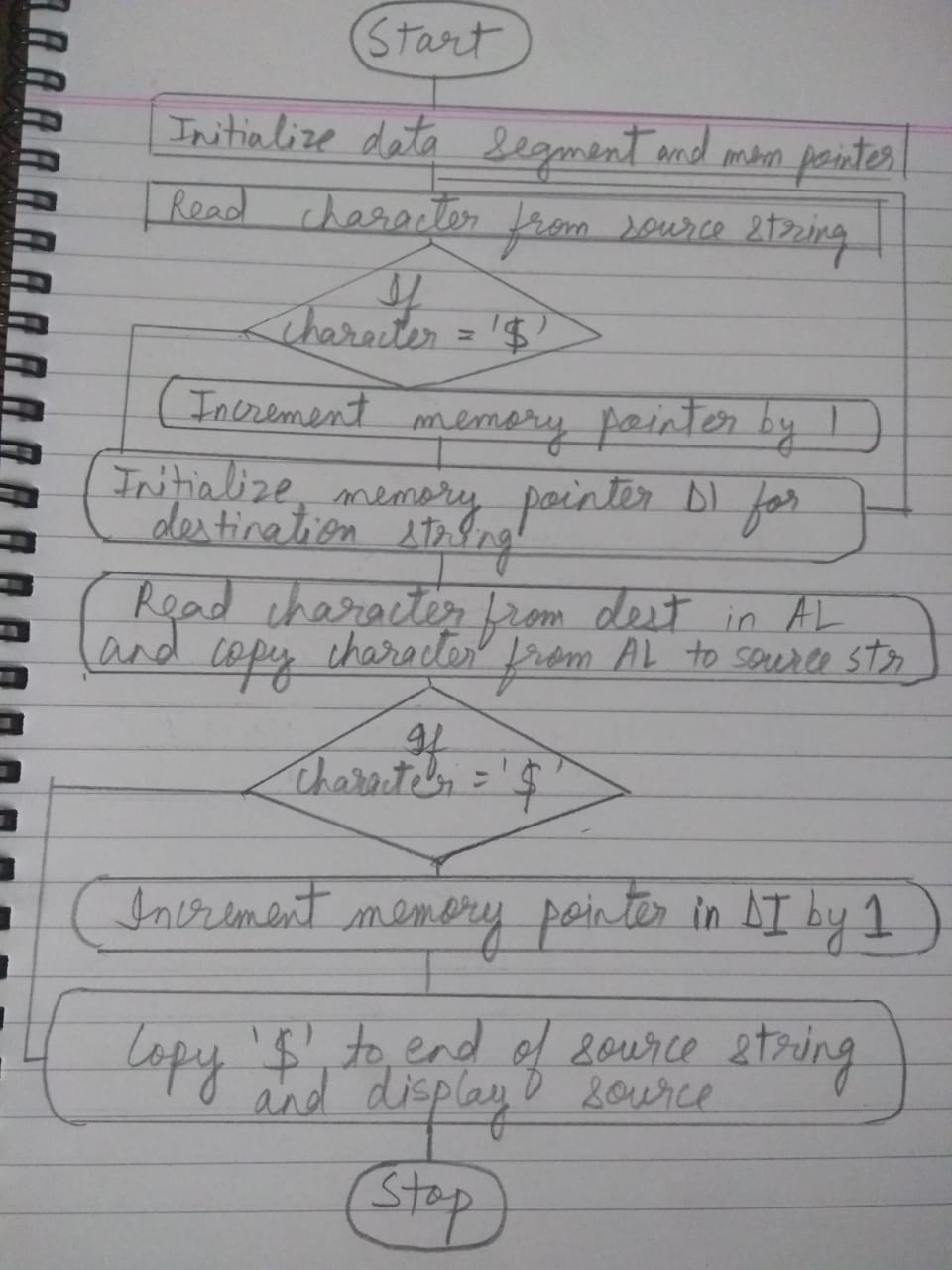
2. Memory pointers source and destination.

3. Move memory of source string to end of string.

4. Copy characters from destination string to source.

5. Stop.

Flowchart:-



**Conclusion**: Hence, we have studied to write an ALP for reversing a string, string length and concatenation of two strings

**ASSIGNMENT 12**

AIM:

Write an ALP to perform conversion from hexadecimal to decimal and decimal to hexadecimal.

THEORY:

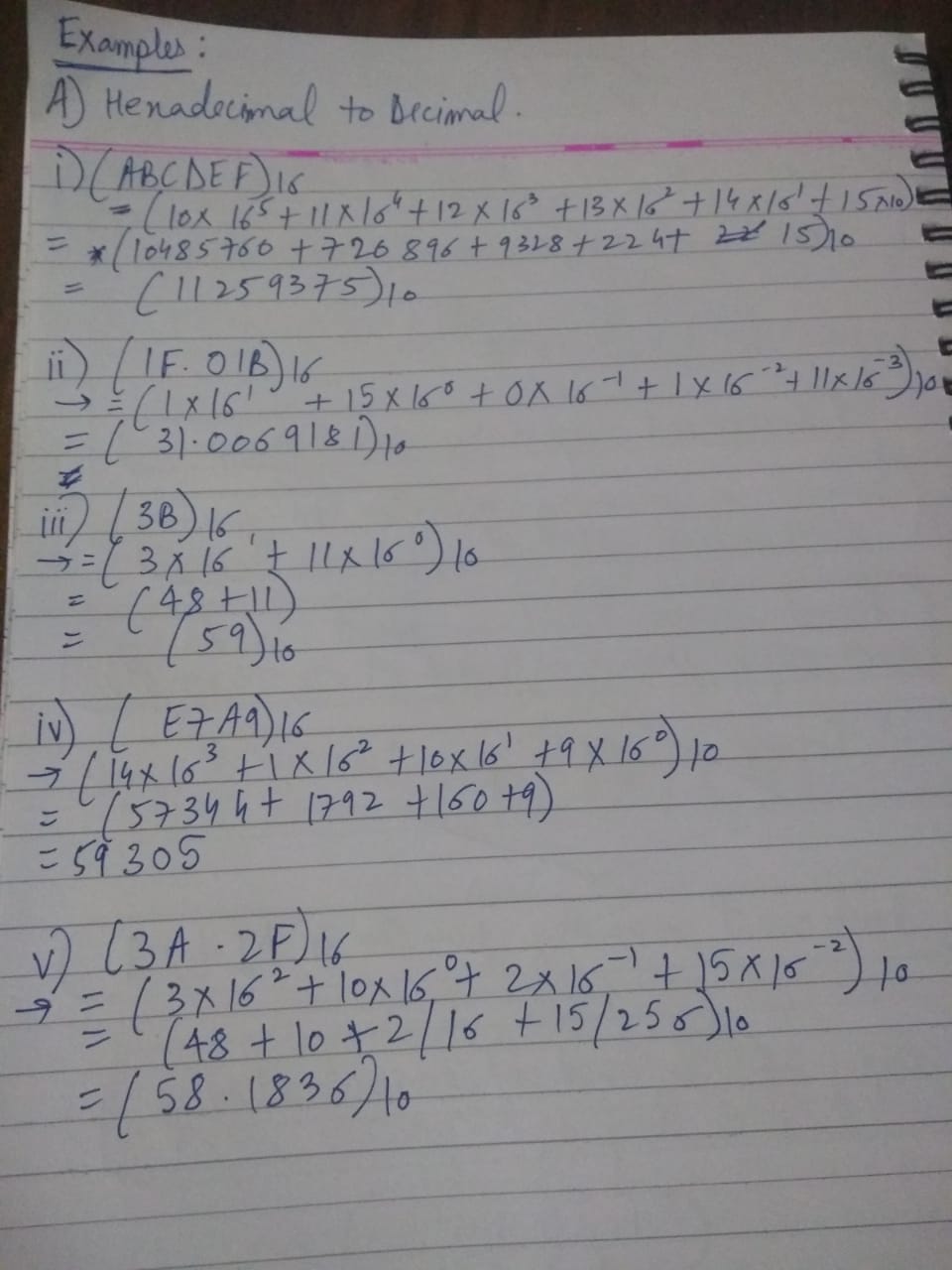
Decimal number system:-

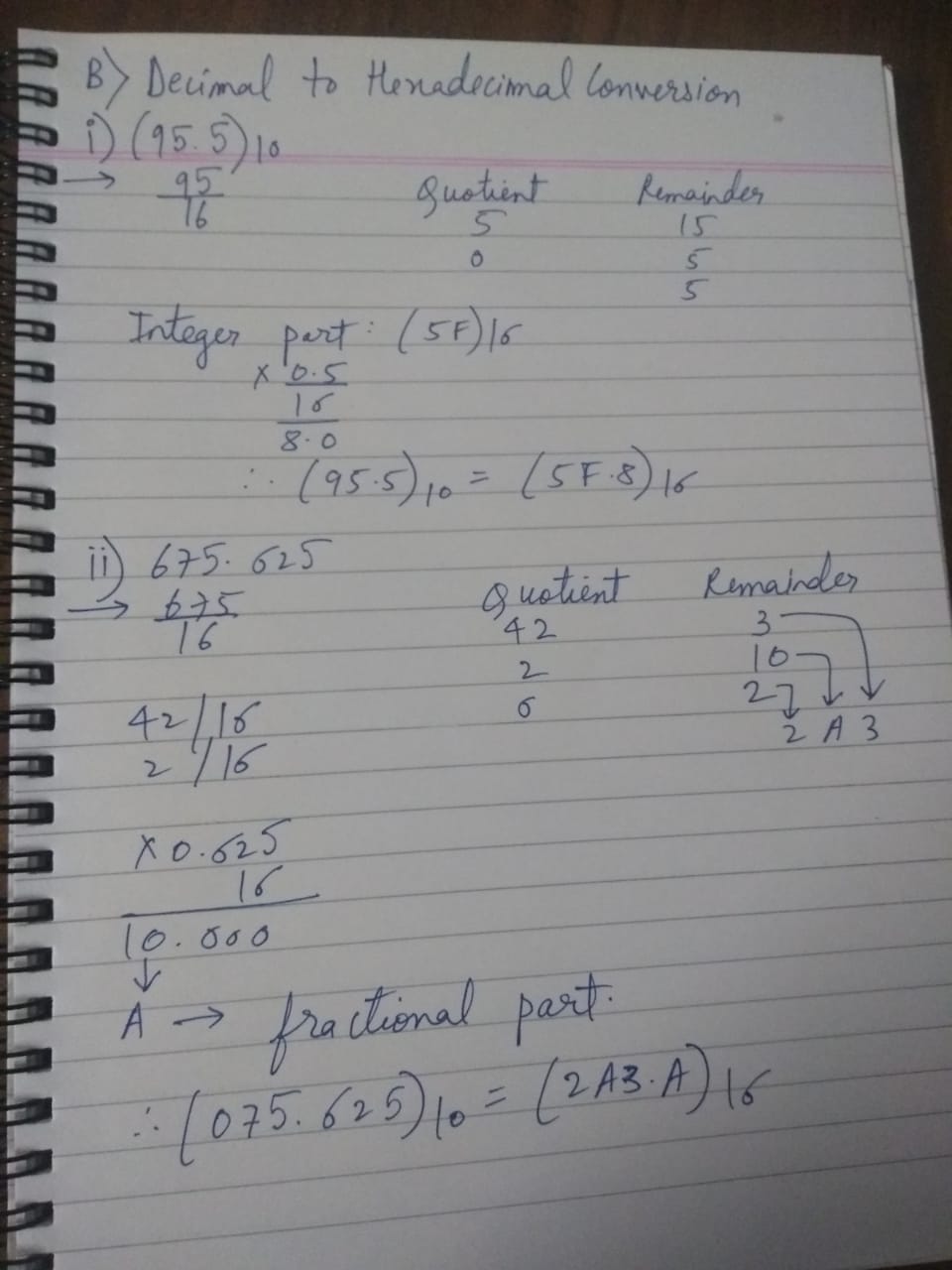
Decimal is a term that describes the base 10 number system, probably the most commonly used number system. The decimal number system consists of 10 digits 0,1, 2,3,4,5,6,7,8. The number after 9 is 10. The number after 19 is 20 and so on. Additional power of 10 requires the addition of another positional digit. The leftmost bit is known as most significant bit MSB and the most rightmost bit is known as least significant bit lsb. Any number of of 0s and 1s can be added to the left of the number without changing the value.

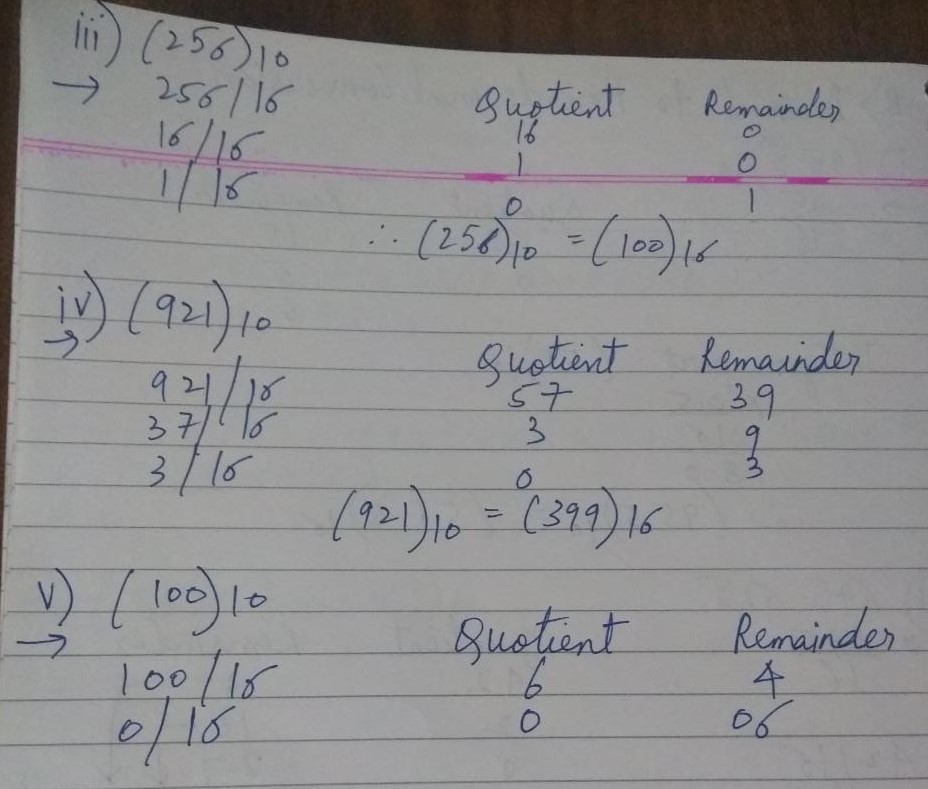
Hexadecimal number system:

The base of numbers system is 16 which requires 16 different symbols to represent numbers. These are numbers 0 to 9 and alphabets from A to F.

Since numeric digits and alphabets both are you represent digits in hexadecimal number system it is an alphanumeric number system.







Algorithm:

A. Decimal to hexadecimal:

1. Load the value stored in the register.

2. Divide the value by 10.

3. Push the reminder to stack.

4. Increase the count.

5. Repeat the step until we get zero in register.

6. Until the count is greater than zero pop from stack.

7. If value at the top of stack is greater than 9.

8. Then add 7 to the value hence we will get hexadecimal character.

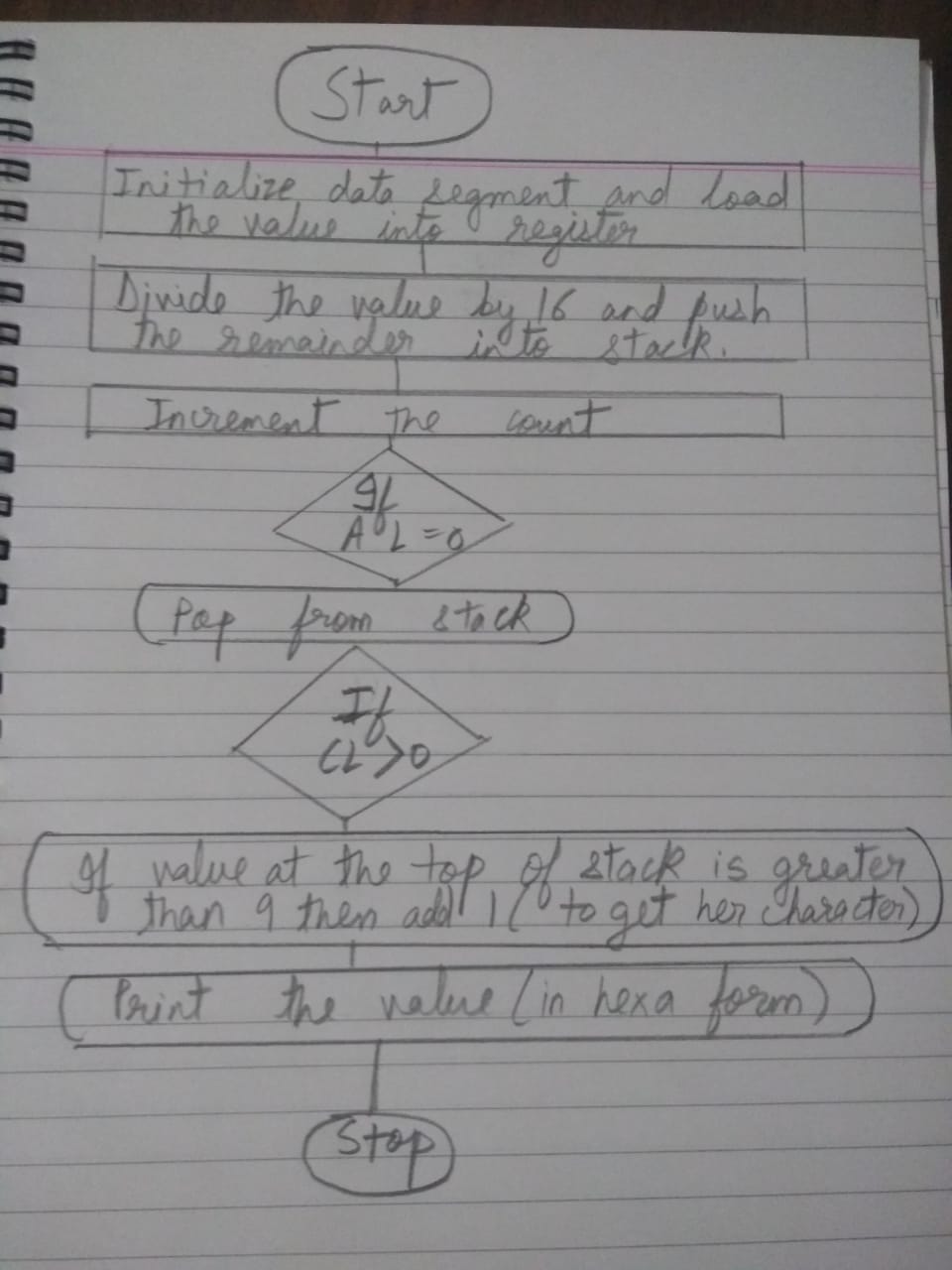
9. Add 48 to the top element to convert it into ASCII.

10. Print character using interrupt.

11. Decrement the count.

12. Stop.

Flowchart:



**Conclusion**:

Hence we have studied to write ALP to convert hexadecimal to decimal and vice versa.