INDEX

PROCEDURE

Start --> Run

CD cobol

1) To Edit
   NE Filename.cob

2) (To Save and Exit F3 --> E)
3) (To save & be in editor F3 --> S)
4) (To save & Quit F3 --> Q)

5) To Compile
   Cobol> Cobol filename.cob;
   No errors or warnings

6) To Run
   Cobol> Runcob Filename.cob
   Enter data

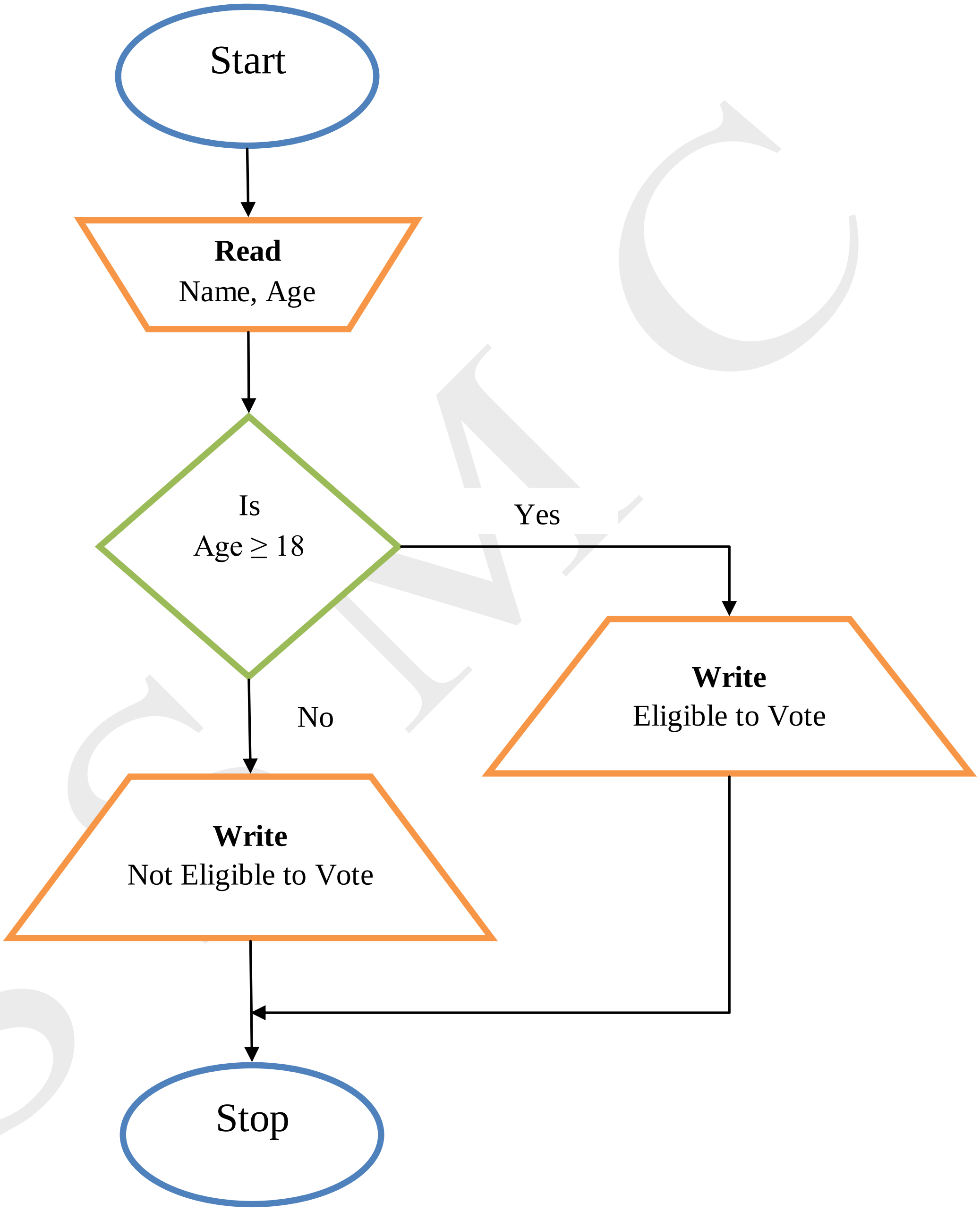
Program No.1

Program in COBOL to check whether a person Eligible for Vote or not.

Algorithm :

1. Execute program
2. Enter Name and Age of a person
3. Check the age of a person
4. If age is equal or greater than 18, then display “Eligible to Vote” Else Display “Not Eligible to Vote”
5. Stop

Flowchart for the program to check whether a person Eligible for Vote or not.



CODE :

```
*PROGRAM TO CHEAK WHEATHER A PERSON ELIGIBLE FOR VOTE OR NOT.
 IDENTIFICATION DIVISION.
 PROGRAM-ID. VOTE.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77 NAME PIC A(7).
 77 AGE PIC 99.
 PROCEDURE DIVISION.
 P1.
     DISPLAY " ENTER THE NAME AND AGE ".
     DISPLAY " ".
     ACCEPT NAME.

     ACCEPT AGE.
     DISPLAY " ".
 P2.
     IF AGE = 18 OR > 18
     DISPLAY "******************************* "
     DISPLAY NAME " IS ELIGIBLE TO VOTE "
     ELSE
     DISPLAY "******************************* "
     DISPLAY NAME " IS NOT ELIGIBLE TO VOTE".
     DISPLAY "******************************* "
     STOP RUN.
```

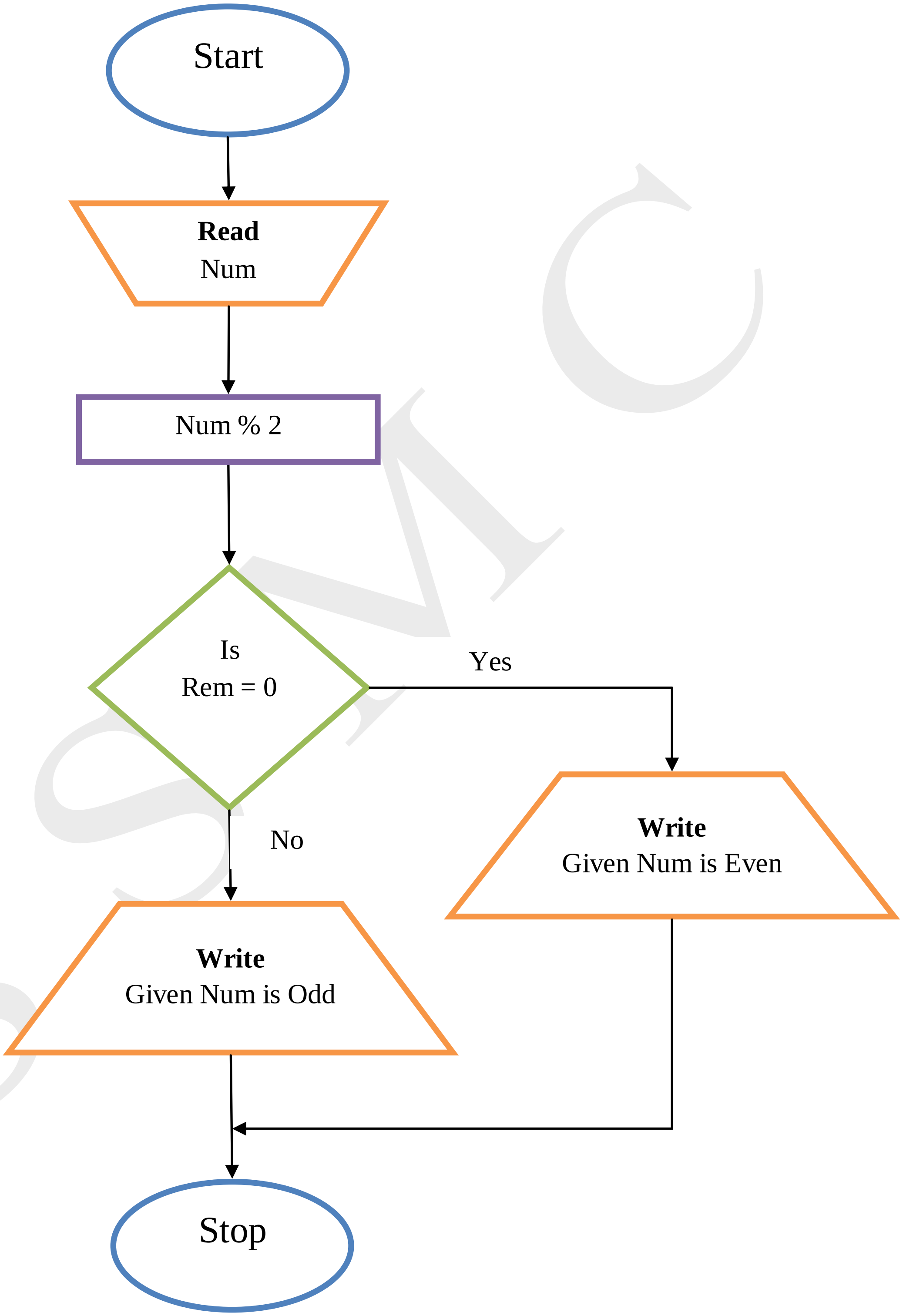
Program No.2

Program in COBOL to find the given number is Even or Odd.

Algorithm :

1. Execute program
2. Enter the number
3. Compute : Number % 2
4. If Reminder = 0, then display “Given Number is Even” Else Display “Given Number is Odd”
5. Stop

Flowchart for the program to find the given number is Even or Odd.



CODE :

```
*PROGRAM TO FIND THE EVEN OR ODD NUMBER.
 IDENTIFICATION DIVISION.
 PROGRAM-ID. EO.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77 N PIC 9(3).
 77 R PIC 9.
 PROCEDURE DIVISION.
 P1.
        DISPLAY " ENTER THE VALUE OF N".
        ACCEPT N.
        DIVIDE N BY 2 GIVING N REMAINDER R.
        IF (R = 0)
        DISPLAY " THE GIVEN NUMBER IS EVEN"
        ELSE
        DISPLAY " THE GIVEN NUMBER IS ODD".
        STOP RUN.
```

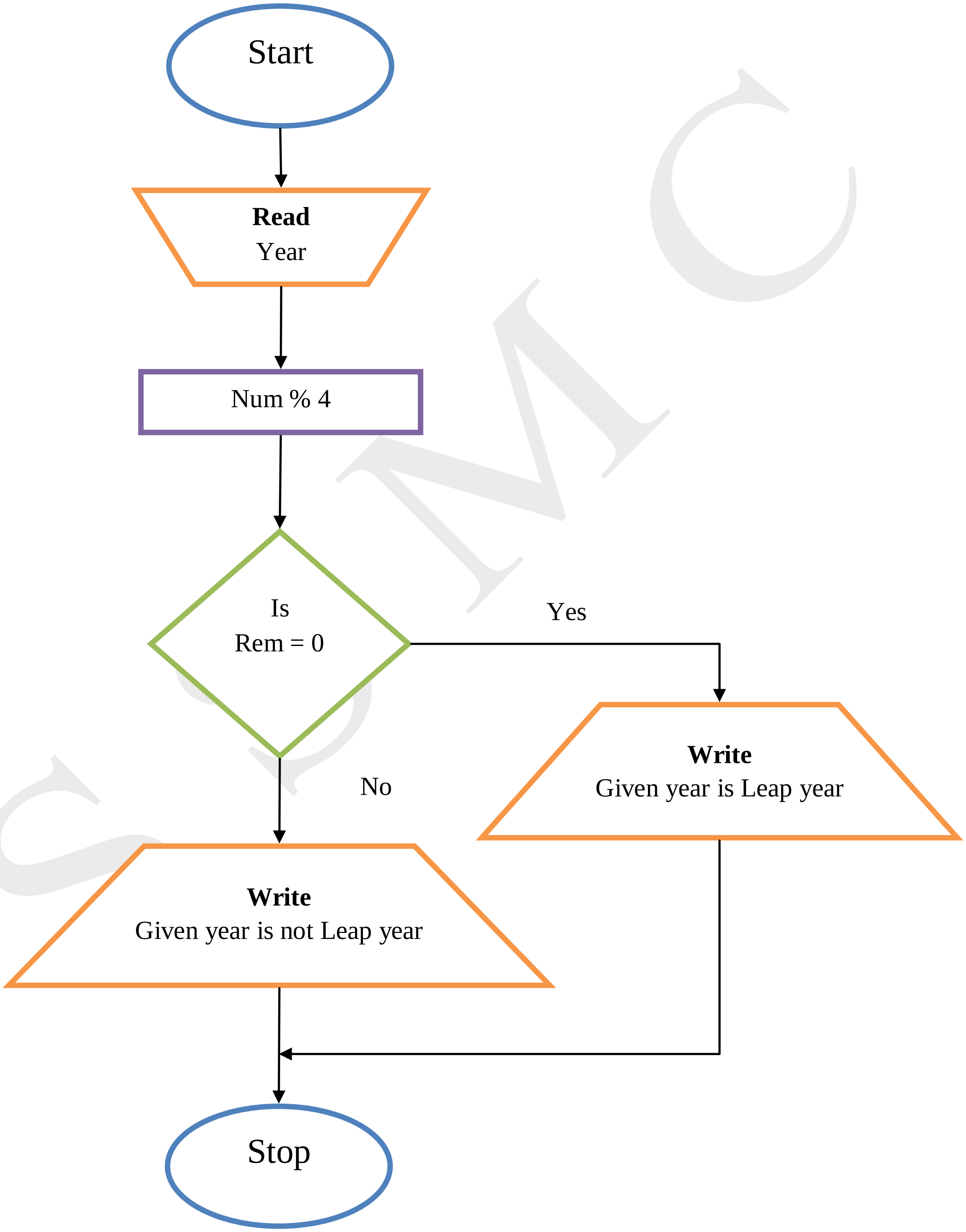
Program No.3

Program in COBOL to find the given year is Leap year or Not.

Algorithm :

1. Execute program
2. Enter the year
3. Compute , Year % 4
4. If Reminder = 0, then display “Given Year is Leap year” Else Display “Given Year is not Leap year”
5. Stop

Flowchart for the program to find the given year is Leap year or Not.



CODE :

```
*PROGRAM TO FIND GIVEN YEAR IS LEAP YEAR OR NOT*
 IDENTIFICATION DIVISION.
 PROGRAM-ID. EVEN.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77 YEAR PIC 9(2).
 77 Q PIC 9(2).
 77 R PIC 9(2).
 PROCEDURE DIVISION.
 P1.
        DISPLAY " ENTER THE YEAR ".
        ACCEPT YEAR.
 P2.
        DIVIDE YEAR BY 4 GIVING Q REMAINDER R.
 P3.
        IF R = 0
        DISPLAY " GIVEN YEAR IS LEAP YEAR "
        ELSE
        DISPLAY " GIVEN YEAR IS NOT LEAP YEAR ".
 P4.
         STOP RUN.
```

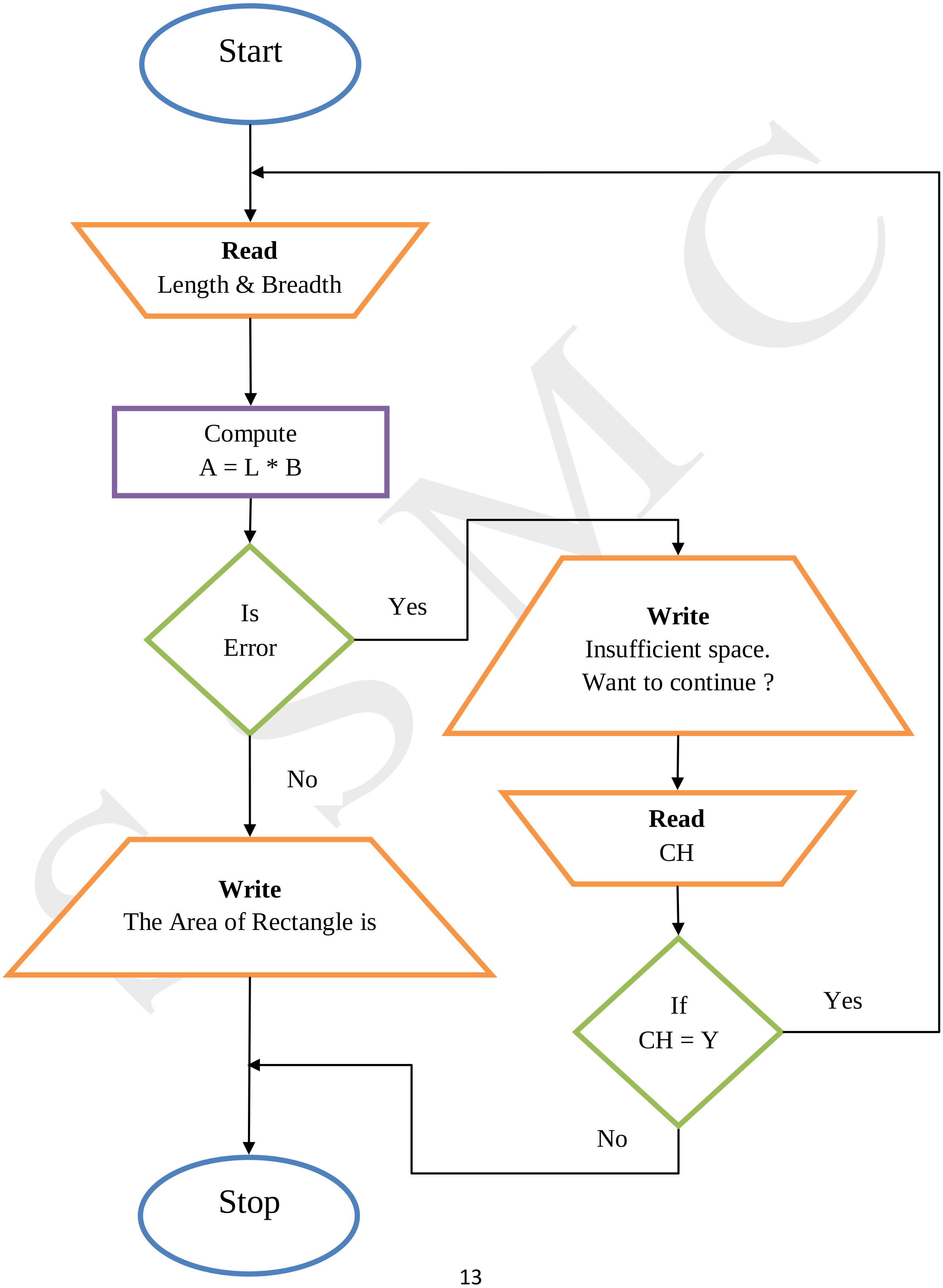
Program No.4

Program in COBOL to find area of rectangle on size of option.

Algorithm :

1. Execute program
2. Enter the Length and Breadth
3. Compute : A = L * B
4. Check size of A
5. If no size error, then display “The area of rectangle is : ”
   Else display “Insufficient space, want to continue ? ”
6. Enter Ch = Y or N
7. If Ch = Y, then repeat step 2
8. Stop

Flowchart for the program to find area of rectangle on size of option.



CODE :

```
*PROGRAM TO FIND AREA OF RECTANGLE ON SIZE OF OPTION
IDENTIFICATION DIVISION.
PROGRAM-ID RECT.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 L PIC 9.
77 B PIC 9.
77 A PIC 9.
77 CH PIC 9.
PROCEDURE DIVISION.
P1.
    DISPLAY " ENTER LENGTH ".
    ACCEPT L.
    DISPLAY " ENTER BREATH ".
    ACCEPT B.
    COMPUTE A = L * B ON SIZE ERROR GO TO P2.
    DISPLAY " AREA = ", A.
    STOP RUN.
P2.
    DISPLAY "INSUFFICIENT SPACE TO STORE RESULT".
    DISPLAY "DO YOU WANT TO CONTINUE".
    ACCEPT CH.
    IF ( CH = 'Y' OR 'y' )
    GO TO P1.
```

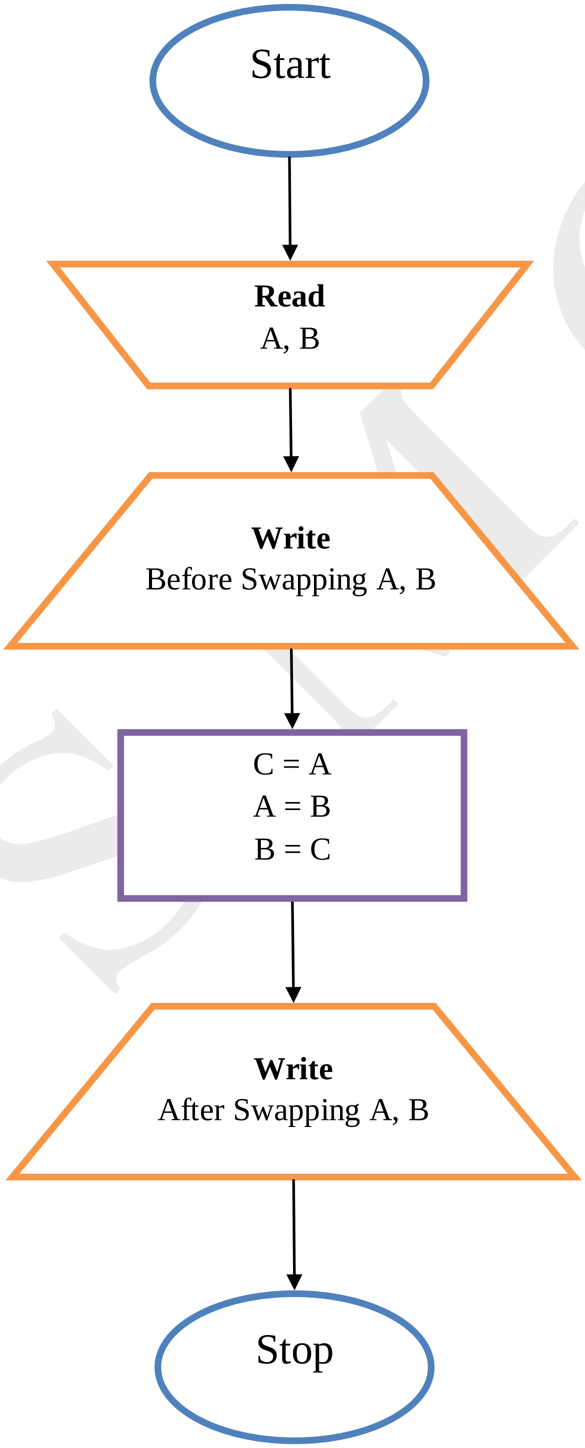
Program No.5

Program in COBOL to swapping numbers.

Algorithm :

1. Execute program
2. Enter the numbers A and B
3. Display numbers A and B before swapping
4. Swap : C = A, A = B, B = C
5. Display numbers A and B after swapping
6. Stop

Flowchart for the program to swapping numbers.



CODE :

```
*PROGRAM TO SWAPPING NUMBERS.
 IDENTIFICATION DIVISION.
 PROGRAM-ID. SWAP.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77 A PIC 9(2).
 77 B PIC 9(2).
 77 C PIC 9(2).
 PROCEDURE DIVISION.
 P1.
        DISPLAY "ENTER THE TWO NUMBER".
        ACCEPT A.
        ACCEPT B.
        DISPLAY "--------------------------------------"
        DISPLAY "BEFORE SWAPPING A = " A "  B = " B "  .
        DISPLAY "-------------------------------------"
        COMPUTE C = A.
        COMPUTE A = B.
        COMPUTE B = C.
        DISPLAY "-------------------------------------"
        DISPLAY "AFTER SWAPPING A = " A "  B = " B " .
        DISPLAY "-------------------------------------"
        STOP RUN.
```

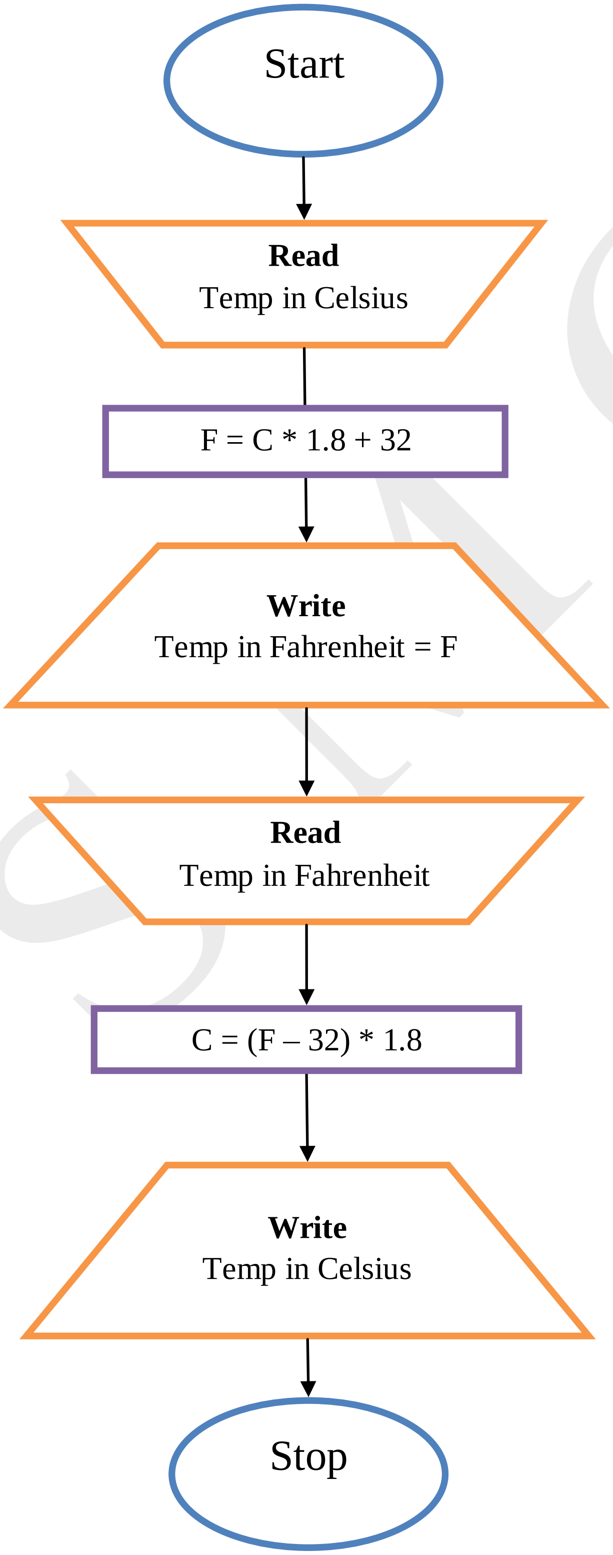
Program No.6

Program in COBOL to find temperature conversion.

Algorithm :

1. Execute program
2. Enter the temperature in Celsius
3. Convert temperature from Celsius to Fahrenheit, F = C * 1.8 + 32
4. Display temperature in Fahrenheit = F
5. Enter the temperature in Fahrenheit
6. Convert temperature from Fahrenheit to Celsius, C = (F – 32) * 1.8
7. Display temperature in Celsius
8. Stop

Flowchart for the program to find temperature conversion.



CODE :

```
*PROGRAM TO FIND TEMPERATURE CONVERSION*
   IDENTIFICATION DIVISION.
   PROGRAM-ID TEMP.
   ENVIRONMENT DIVISION.
   DATA DIVISION.
   WORKING-STORAGE SECTION.
   01 F PIC 9(3).
   01 C PIC 9(3).
   PROCEDURE DIVISION.
   P1.
        DISPLAY " ENTER THE TEMPERATURE IN CELCIUS ".
        DISPLAY " ".
        ACCEPT C.
        COMPUTE F = C * 1.8 + 32.
        DISPLAY " ".
        DISPLAY "======================================== ".
        DISPLAY " TEMPERATURE IN  FAREHNIET   IS : " F.
        DISPLAY "======================================== ".
        DISPLAY " ".
        DISPLAY " ".
        DISPLAY " ENTER THE TEMPERATURE IN FAREHNIET".
        DISPLAY " ".
        ACCEPT F.
        COMPUTE C = (F - 32) / 1.8.
        DISPLAY " ".
        DISPLAY "======================================== ".
        DISPLAY " TEMPERATURE IN  CELCIUS     IS : " C.
        DISPLAY "======================================== ".
        STOP RUN.
```

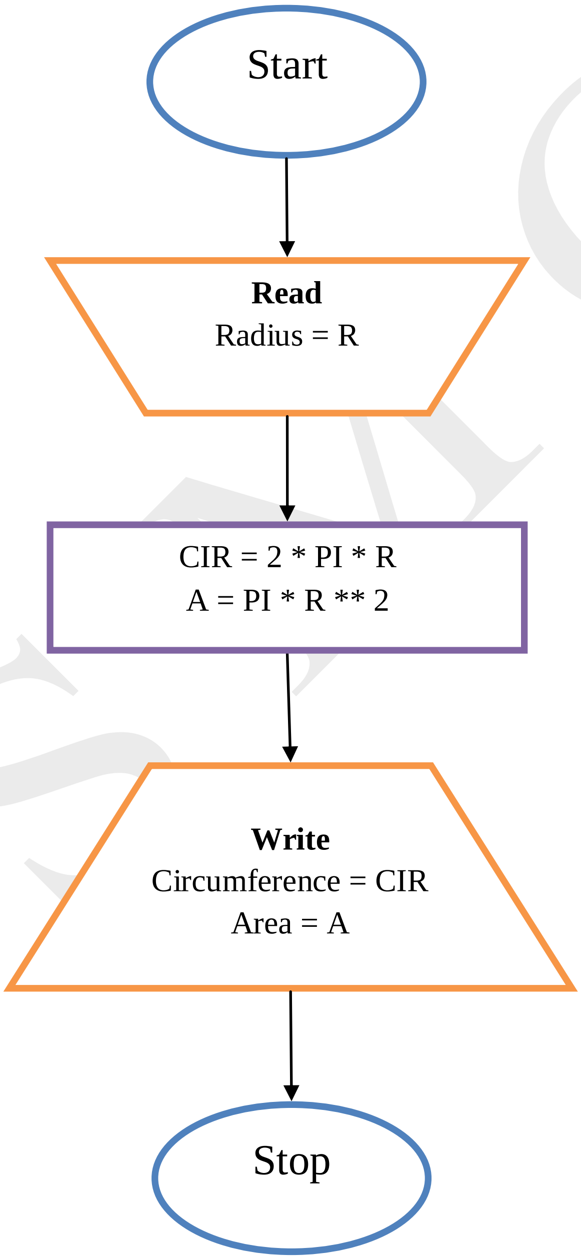
Program No.7

Program in COBOL to find Area and Circumference of a circle.

Algorithm :

1. Execute program
2. Enter the radius of a circle
3. Compute , (i) CIR = 2 * PI * R, (ii) A = PI * R ** 2
4. Display Circumference of a circle
5. Display Area of a circle
6. Stop

Flowchart for the program to find Area and Circumference of a circle.



CODE :

```
*AREA AND CIRCUMFERANCE OF A CIRCLE*
 IDENTIFICATION DIVISION.
 PROGRAM-ID. CIRC.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77 R PIC 9(2)V99.
 77 CIR PIC Z(4).999.
 77 A PIC Z(4).999.
 77 PI PIC 9(1)V999 VALUE 3.142
 PROCEDURE DIVISION.
 PARA1.
          DISPLAY " ENTER THE RADIUS ".
         ACCEPT R.
         COMPUTE CIR = 2 * PI * R.
         COMPUTE A = PI * R ** 2.
         DISPLAY " ".
         DISPLAY " CIRCUMFERANCE = ", CIR.
         DISPLAY " AREA          = ", A.
         STOP RUN.
```

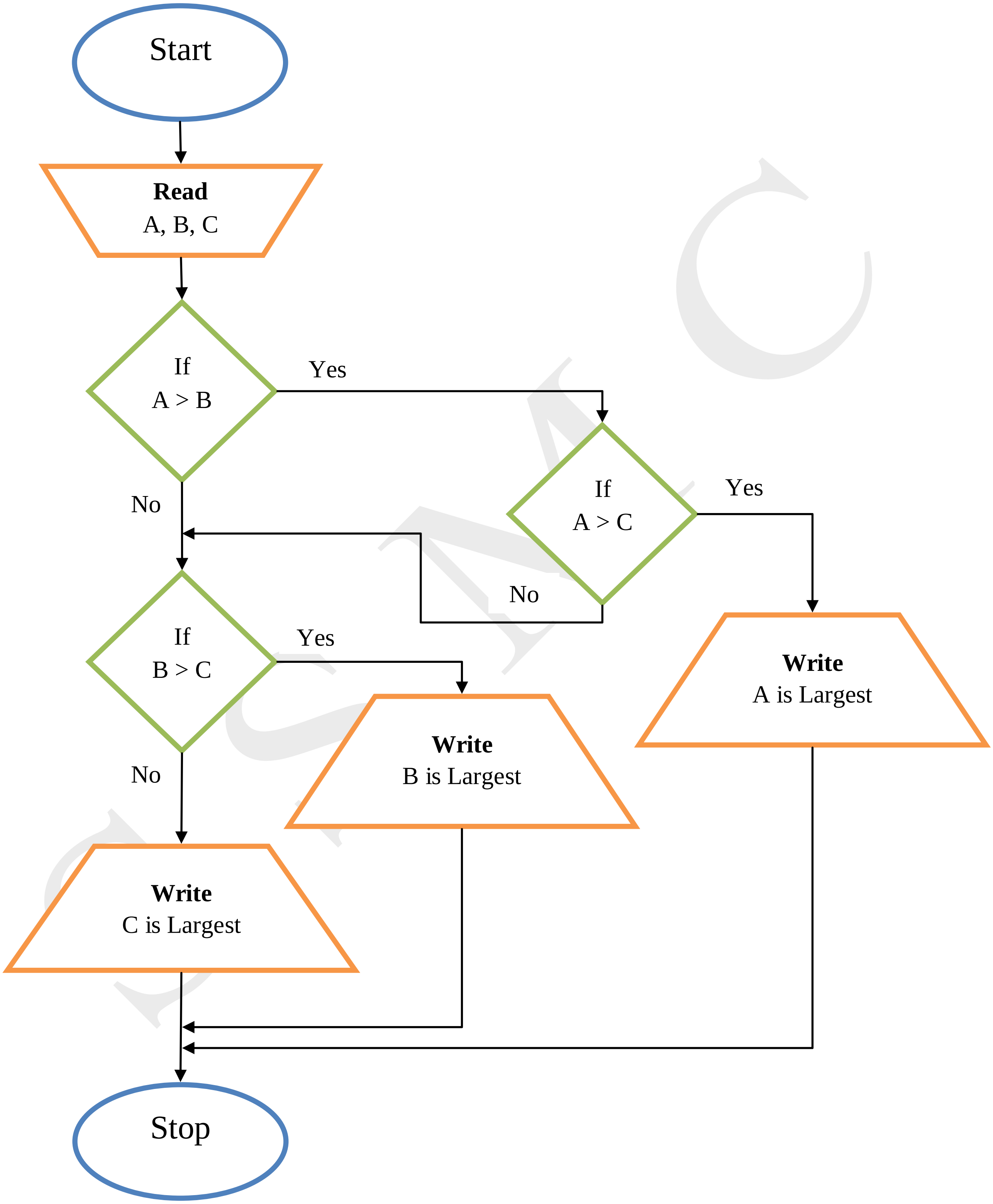
Program No.8

Program in COBOL to find Largest of Three numbers.

Algorithm :

1. Execute program
2. Enter three numbers A,B,C
3. Check conditions :
   a) If (A > B) and (A > C), Then Display A is Largest
   b) If (A < B) and (B > C), Then Display B is Largest
   c) If (A < B) and (B < C), Then Display C is Largest
4. Stop

Flowchart for program to find Largest of Three numbers.



CODE :

```
*PROGRAN TO FIND LARGEST OF THREE NUMBERS
 IDENTIFICATION DIVISION.
 PROGRAM-ID. BIG.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77 A PIC 9(2).
 77 B PIC 9(2).
 77 C PIC 9(2).
 PROCEDURE DIVISION.
 P1.
        DISPLAY "ENTER THE THREE NUMBERS".
        DISPLAY " ".
        ACCEPT A.
        ACCEPT B.
        ACCEPT C.
        IF ( A > B )
        IF ( A > C )
        DISPLAY "========================="
        DISPLAY A " IS LARGEST NUMBER "
        DISPLAY "========================="
        ELSE
        DISPLAY "========================="
        DISPLAY C " IS LARGEST NUMBER "
        DISPLAY "========================="
        ELSE
        IF ( B > C )
        DISPLAY "========================="
        DISPLAY B " IS LARGEST NUMBER "
        DISPLAY "========================="
        ELSE
        DISPLAY "========================="
        DISPLAY C " IS LARGEST NUMBER "
        DISPLAY "========================="
        STOP RUN.
```

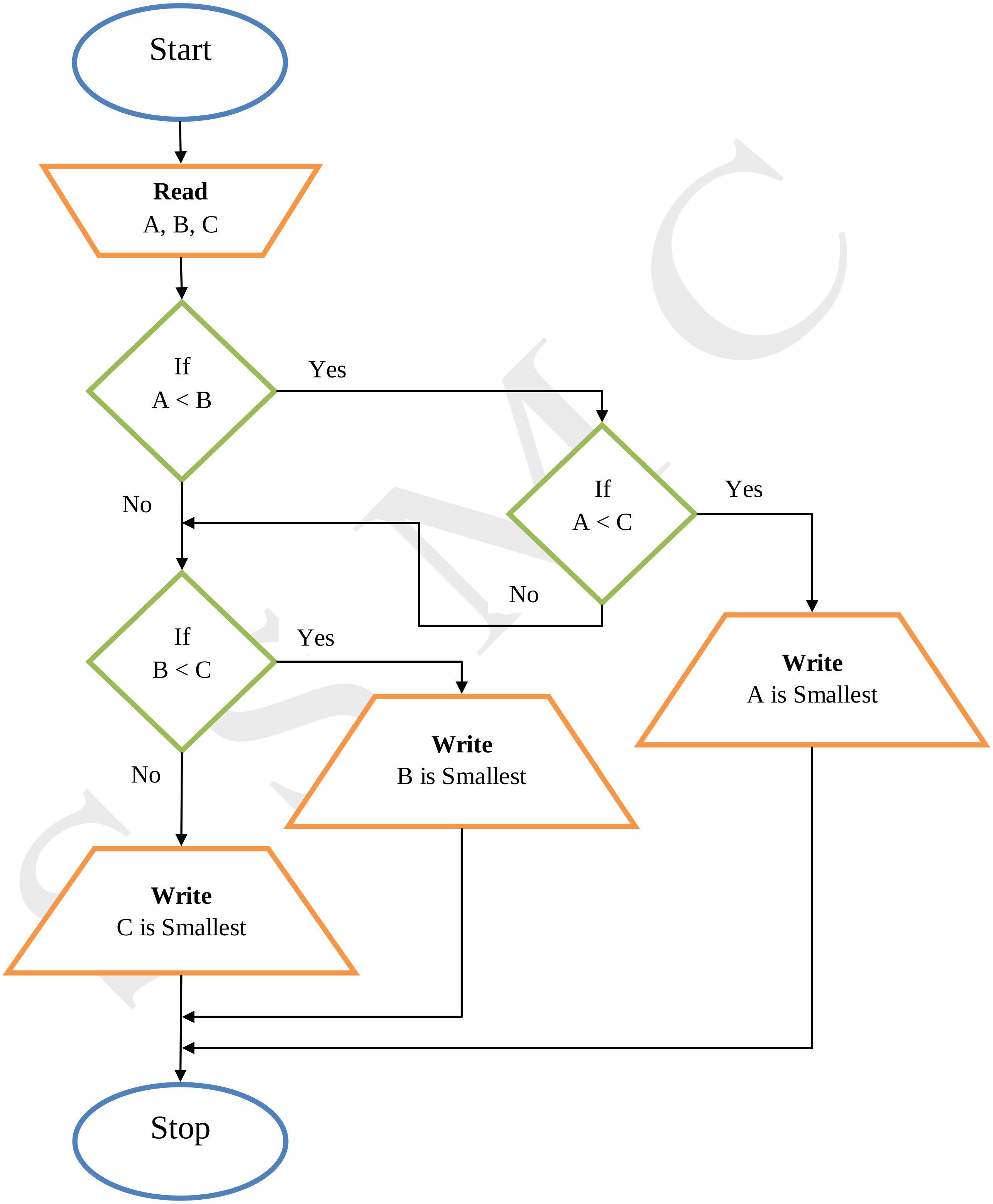
Program No.9

Program in COBOL to find Smallest of Three numbers.

Algorithm :

1. Execute program
2. Enter three numbers A,B,C
3. Check conditions :
    a) If (A < B) and (A < C), Then Display A is Smallest
    b) If (A > B) and (B < C), Then Display B is Smallest
    c) If (A > B) and (B > C), Then Display C is Smallest
4. Stop

Flowchart for program to find Smallest of Three numbers.



CODE :

```
*PROGRAN TO FIND LARGEST OF THREE NUMBERS
    IDENTIFICATION DIVISION.
    PROGRAM-ID. BIG.
    ENVIRONMENT DIVISION.
    DATA DIVISION.
    WORKING-STORAGE SECTION.
    77 A PIC 9(2).
    77 B PIC 9(2).
    77 C PIC 9(2).
    PROCEDURE DIVISION.
    P1.
         DISPLAY "ENTER THE THREE NUMBERS".
         DISPLAY " ".
         ACCEPT A.
         ACCEPT B.
         ACCEPT C.
         IF ( A < B )
         IF ( A < C )
         DISPLAY "========================"
         DISPLAY A " IS SMALLEST NUMBER "
         DISPLAY "========================"
         ELSE
         DISPLAY "========================"
         DISPLAY C " IS SMALLEST NUMBER "
         DISPLAY "========================"
         ELSE
         IF ( B < C )
         DISPLAY "========================"
         DISPLAY B " IS SMALLEST NUMBER "
         DISPLAY "========================"
         ELSE
         DISPLAY "========================"
         DISPLAY C " IS SMALLEST NUMBER "
         DISPLAY "========================"
         STOP RUN.
```

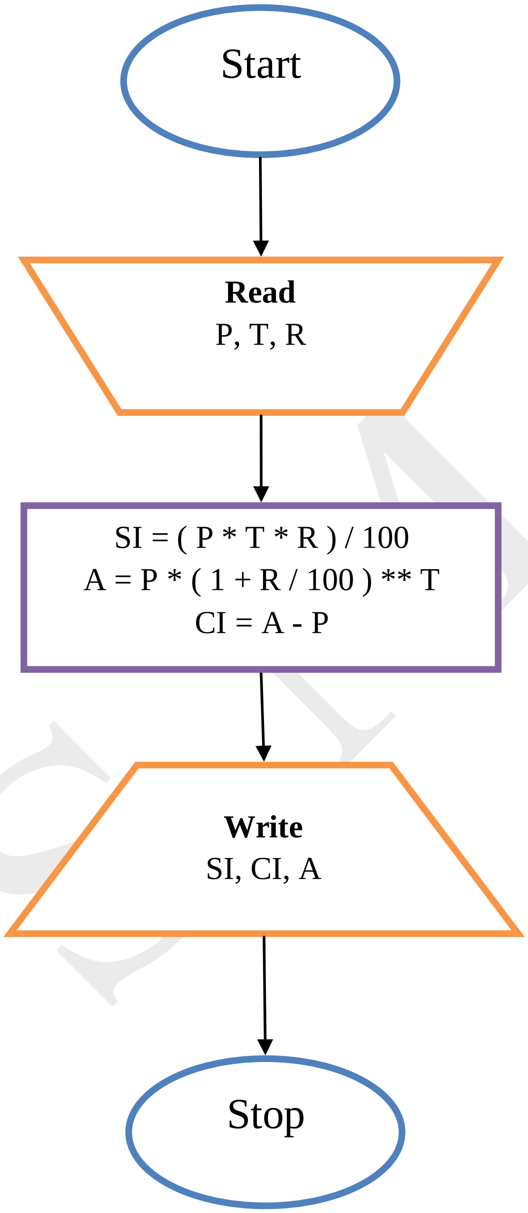
Program No.10

Program in COBOL to calculate simple and compound interest.

Algorithm :

```
1. Execute program
2. Enter :
   a) Principle amount = A
   b) Period = T
   c) Rate of interest = R
3. Compute :
   a) $SI = ( P * T * R ) / 100$
   b) $A = P * ( 1 + R / 100 ) ** T$
   c) $CI = A - P$
4. Display “Simple Interest = “, SI
5. Display “Compound Interest = “, CI
6. Display “Amount = “, A
7. Stop
```

Flowchart for program to calculate simple and compound interest.



CODE :

```
*PROGRAM TO CALCULATE SIMPLE AND COMPOUND INTERST
 IDENTIFICATION DIVISION.
 PROGRAM-ID. FFF.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 01 P PIC 9(4).
 01 T PIC 99.
 01 R PIC 999V99.
 01 SI PIC 999.99.
 01 CI PIC 999.99.
 01 A PIC 9999V99.
 PROCEDURE DIVISION.
 P1.
      DISPLAY " ENTER THE VALUE OF PRINCIPLE AMOUNT ".
      ACCEPT P.
      DISPLAY "ENTER THE VALUE OF PRIODE ".
      ACCEPT T.
      DISPLAY " ENTER THE VALUE OF RATE OF INTERST ".
      ACCEPT R.
 P2.
      COMPUTE SI = ( P * T * R ) /100.
      COMPUTE A = P * ( 1 + R / 100 ) ** T.
      COMPUTE CI = A - P.
 P3.
      DISPLAY " SIMPLE INTERST = ", SI.
      DISPLAY " COMPOUND INTERST = " , CI.
      DISPLAY " AMOUNT = ", A.
      STOP RUN.
```

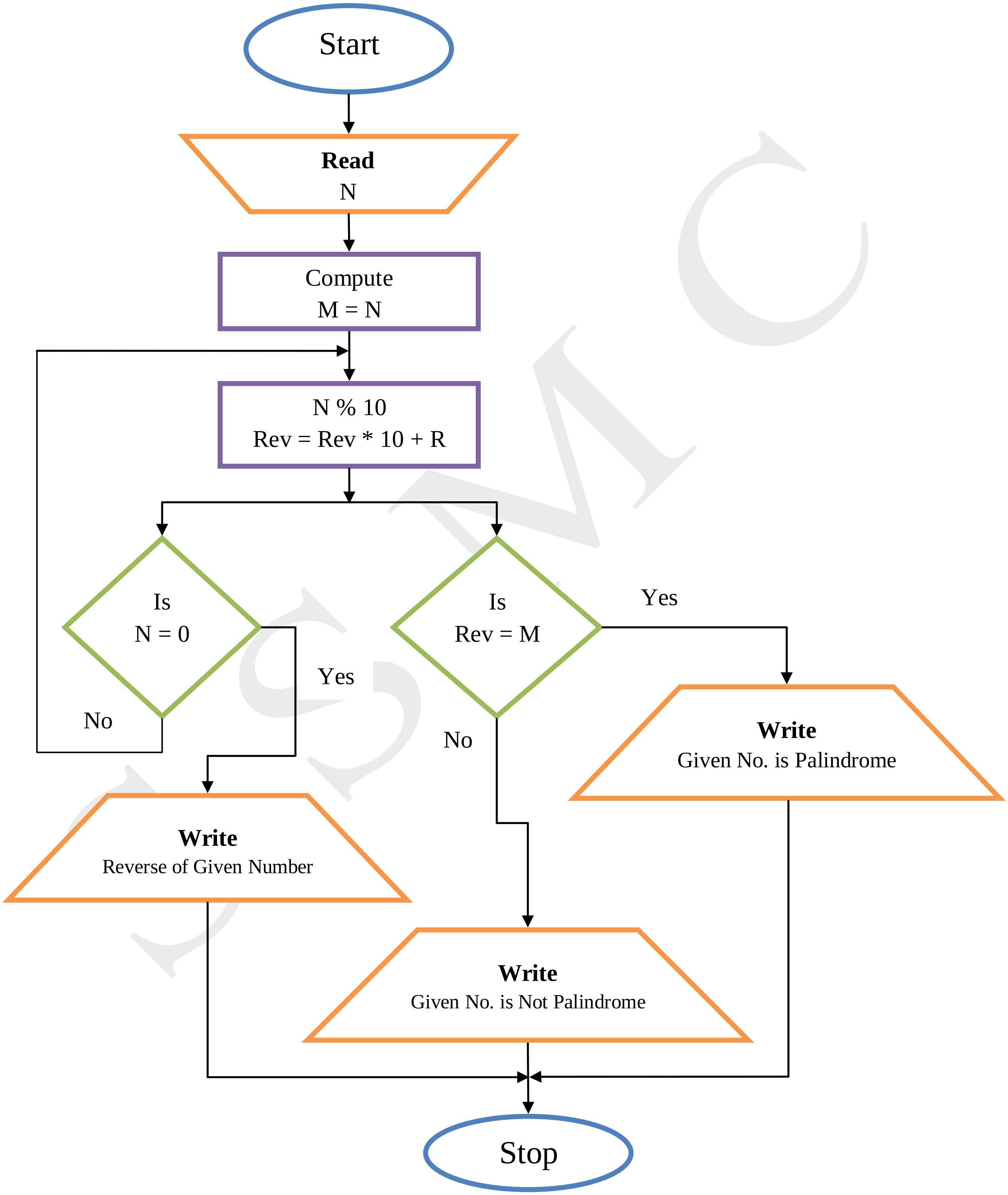
Program No.11

Program in COBOL to find reverse of number and check for palindrome.

Algorithm :

1. Execute program
2. Enter the number to find reverse, N
3. Compute, M = N
4. Divide N by 10 giving N reminder R
5. Compute Rev = Rev * 10 + R
6. If N = 0, then Move Rev to Rev1
7. Display “Reverse of given number = “, Rev1
8. Else go to step 4
9. If Rev = M
10. Display “ Given number is Palindrome”
11. Else Display “Given number is not Palindrome”
12. Stop

Flowchart for program to find reverse of number and check for palindrome.



CODE :

```
*PROGRAM TO FIND REVERSE OF NUMBER AND CHECK FOR PALINDROME
 IDENTIFICATION DIVISION.
 PROGRAM-ID. JJ.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77 N PIC 9(7).
 77 R PIC 9(7).
 77 REV PIC 9(7) VALUE 0.
 77 REV1 PIC Z(7).
 77 M PIC 9(7).
 PROCEDURE DIVISION.
 P1.
         DISPLAY " ENTER THE NUMBER TO FIND REVERSE ".
         ACCEPT N.
         COMPUTE M = N.
 P2.
         DIVIDE N BY 10 GIVING N REMAINDER R.
         COMPUTE REV = REV * 10 + R.
         IF N = 0
         MOVE REV TO REV1
         DISPLAY " REVERSE OF GIVEN NUMBER = " , REV1
         ELSE
         GO TO P2.
         IF REV = M
         DISPLAY " GIVEN NUMBER IS PALINDROME "
         ELSE
         DISPLAY " GIVEN NUMBER IS NOT PALINDROME ".
         STOP RUN.
```