Vaida - Practical 1

Aim:- Write a C program to understand the same data types

Algorithm:-

1. Declare a variable integer to store roll no., a float variable to store percentage & two string arrays for storing name and mobile no.
2. Print using the printf() method to ask for values and use scanf() to get the required values.
3. After receiving the values, print them one by one using the printf() statement.

Conclusion:-

Thus, the integer, character and float data-types have been studied.

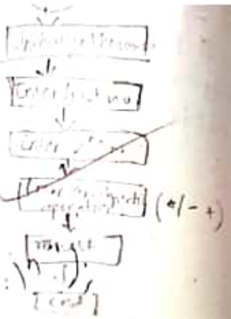
code :-

```
 #include <stdio.h>
#  include <conio.h>.
void main ()
{   int roll-no ;
    char name [10], mobile_no[10];
    float percentage;
    clrscr ();
    printf ("Enter student's name : \n");
    scanf ("%s", &name);
    printf ("Enter student's rollno : \n"),
    scanf ("%d", &roll no);
    printf ("Enter student's mobile no. \n").
    scanf ("%s", & mobile_no);
    printf ("Enter student's percentage : \n");
    scanf ("%f", &percentage);
    printf ("student's name : %s\n", name);
    printf ("student's roll no : %d\n", roll.no)
    printf ("student's mobile no: %s\n",
                          mobile_no);
    printf ("student's percentage : %f\n" percentage)
    getch();
}
```

Output:-

```
Enter student's name:
Vaidehi
Enter student's roll no :
1861
Enter student's percentage :
89.12
Enter student's mobile no :
99999557
                         student's percentage : 89.12
```

```
q7 code :-
//dynamic calculator.
#include <stdio.h>
. # include <conio.h>
void main()
{
        int num1, num2;
        float add, sub, mult, div,
        clrscr();
        printf("Enter first numbers: \n");
        scanf("%d", &num1);
        printf("Enter second numbers . \n");
        scanf("%d", &num2);
        add = num1+num2;
        sub = num1-num2;
        mult = num1 * num2,
        div = num1/num2,
        printf("Addition of %d and %d is %f\n",
                num1, num2, add);
        printf("Subtraction of %d and %d is %f\n",
                num1, num2, sub);
        printf("Multiplication of %d and %d is %f\n",
                num1, num2, mult);
        printf("Division of %d and %d is %f\n",
                num1, num2, div);
        getch();
}.
```



Practical 2 :-.

Aim :- Write C program on operator & expression.

Write a program to create a dynamic calculator.

Algorithm:

Step1: Declare a variable name for first & second number as integer.

Step2: Now use scanf f^n to receive input from user

Step3:- Now to add 2 no.s given by user, use the expression num1 + num2.

Step4:- Now to sub two no.s given by use, use expression num1 - num2.

Step5. - Again use expression num1 * num2 if user wishes to multiply the two inputs.

Step6. - Use expression num1/num2 if user wishes to divide the two inputs.

Step7:- Now use printf function to display output.

b) Write a program in C to explain ternary operator.

Algorithm :-

Step1:- Declare variables a, b & x as integers .

Step2:- Store the value of a as 5 & Store the value of b as 15

Step3:- Now to compare b/w who is greater use ternary operator x to find .

Step4:- Use printf function to display output .

Conclusion:-

These programs help us in having better understanding about operators & expressions.

Nirmal

Output -

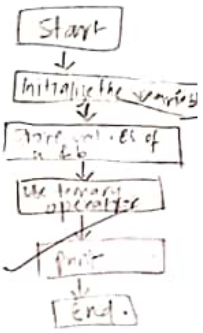
```
Enter first no = 4
Enter second no = 4
Addition of 4 & 4 is 8
Subtraction of 4 and 4 is 0 .
Multiplication of 4 and 4 is 16 .
Division of 4 and 4 is 1 .
```

b) code :-

```
// ternary operator
#include <stdio.h>
#include <conio.h>
void main()
{       int a, b, x;
        clrscr();
        a = 5 ;
        b = 15;
        x = (a > b) ? a : b;
        printf(x);
         getch();
}
```

output

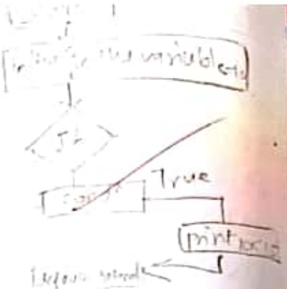
15



code :-

```
// if statement .
# include <stdio.h>
# include <conio.h>
void main()
{ int i=10;
  clrscr();
  if (i>15)
  { printf ("10 is less than 15\n");
  }
  printf ("I am not in if \n");
  getch();

}
```

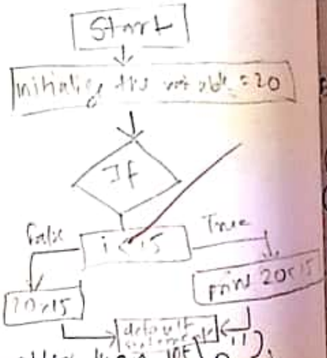


Output :-

```
I am not in if .
```

b) code :-

```
// if else statement .
# Include <stdio.h>
# include <conio.h>.
void main()
{     int i=20;
      clrscr();
      if (i<15).
   { print ("20 is smaller than 15\n");
   }
   else :
   {   printf ("20 is greater than 15\n");
   }
```



Practical 3 -

Aim :-
Write a program in C on decision statement (if, if else, nested if).

a) Write a program in C to explain if statement .

Algorithm :-

1. Declare a variable as integer and assign its i.e. 20.
2. Now to compare whether 20 is greater than if statement.
3. If the condn is true, printf that 20 is less t if condn is false skip the if statement & p I a not in if.

b) Write a program in C to explain if else stat

1. Declare a variable as integer & assign its va
2. Now to compare the given value if its gr a not use if else condnal statement.
3. If condn is true the print 20 is less than 1 condn is false then print 20 is greater than

c) Write a program in C to explain nested if statement

Algorithm :-

① Declare a variable as integer and assign value i=20
② Now use nested if logic to compare if given no is greater or not.
③ If first condn is true then go to second condn. If 2nd condn is also true then print that 20 is greater than 15 & 12. If one of the condn's are not true then skip the part & print 20 is greater than 15 & 12.

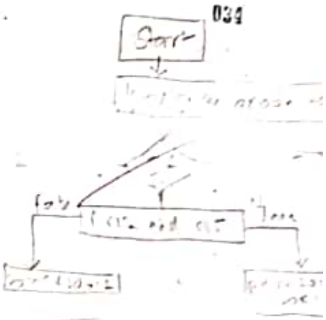
Conclusion :-

These programs help us to understand the working of if, if else & nested if conditional statement.

Correct

Output :-

20 is greater than 15.



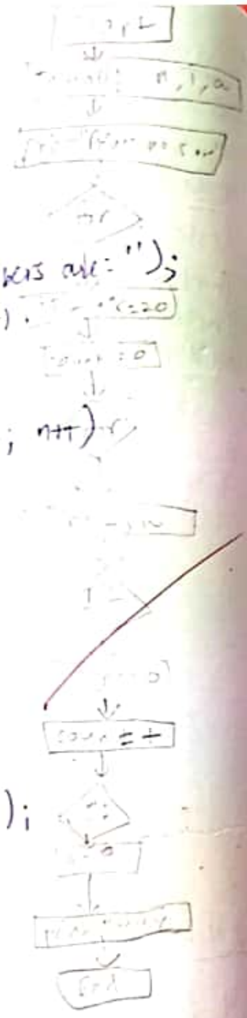
i) code :-

```
# nested if
# include <stdio.h>
# include <conio.h>
void main()
{
        int i = 20,
        clrscr();
        if (i<15):
        {   if (i<12).
            { printf (" 20 is less than 15 &1
            }
        }
         else :
         { printf (" 20 is greater than 15 &1
           }
    getch();
  }.
```

Output :-

20 is greater than 10 & 12.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int n,i,a;
  clrscr();
  printf("The prime numbers are:-");
  for(i=2;i<=20;i++).
  {  a=0;
     for(n=2;n<i;n++)
     {  if(i%n==0).
        {
          a++;
        }
     }
     if(a==0)
     {
       printf("%d\n",i);
     }
  }
  getch();
}
```



Output :- The prime no.s are:- 2,3,5,7,11,13,17,19

Practical 4:-

(i) Aim:- To display prime no.s using for loop.

Algorithm:-

1) Initialize three variables out of which 2 are loop variables & ~~are~~ 1 is a count variable.

2) Initialize a for loop from 2 to 20 (values for the loop variable). Let the count variable be 0.

3) Next another loop within the loop in step 2 that goes from 2 to the first loop variable 2.

4) Use the if conditional statement to check whether (1st loop variable % 2nd loop variable ==0). If true, increment count variable by 1.

5) Come out of the second loop & check whether the count variable is 0. If true, print the no. (1st loop variable).

6) Terminate the program.

Conclusion:- Prime no.s were displayed using for loop

(1) Write a C program on fibonacci series.

Aim:- To write a program on fibonacci series.

Algorithm:-

(1) Start the Turbo C.
(2) Declare the variables n1, n2, n3, i, no.
(3) Initialize the variable n1=0, n2=1, no=0.
(4) Enter the no. of terms of fibonacci series to be printed.
(5) Print first 2 ~~no.s~~ term of series as n1 = 0 & n2=1.
(6) Use the for loop as per following step:-

```
n3 = n1+n2
n1 = n2
n2 = n3
```

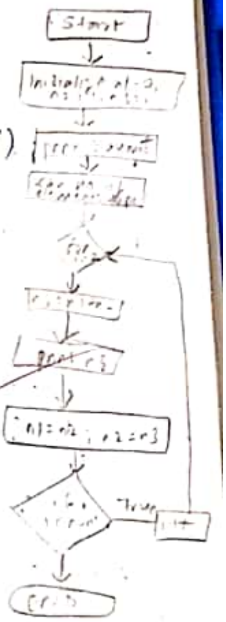
(7) Increase the value of i elements each time by 1.
(8) Print the value of no.s.
(9) End the program.

Conclusion:-
Thus, we have successfully execute fibonacci series on Turbo C.

```
#include <stdio.h>
#include <conio.h>
void main()
{
 int n1=0, n2=1, n3, i, number;
 clrscr();
 printf("Enter number of elements\n")
 scanf("%d", &number);
 printf("%d %d", n1, n2);
 for(i=2; i < number; i++)
 {
   n3 = n1 + n2;
   printf("%d", n3);
   n1 = n2;
  n2 = n3;
  }
  getch();
 }
```

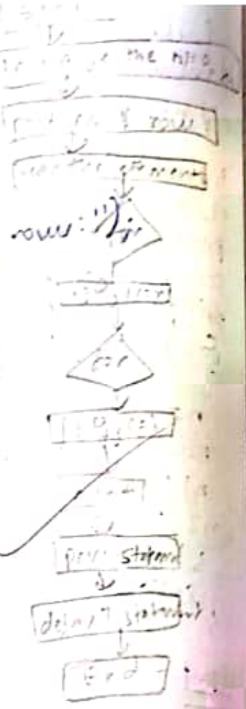
Output:-

```
Enter number of elements:-
 8
 0 11235813
```



No.

```
#include<stdio.h>
#include<conio.h>
void main()
{ int n1=0, r, i, j;
  clrscr();
  printf(" Enter the number of rows:");
  scanf("%d",&r);
  printf("\n");
  for(i=0; i<=r; i++)
  {
   for(j=0; j<=i; j++)
   { n1++;
     printf("%d", n1);
   }
   printf("\n");
  }
  getch();
```



Output:-

```
Enter the no. of rows : 4
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Aim:-

① Write a C program on following expression :-

```
1
2  3
4  5  6
7  8  9  10
11 12 13 14 15
```

Step1:- Start Turbo C .

Step2:- Declare variables as rows, i, j, number = 1

Step3:- Display the no. of rows .

Step4:- Enter the for loop as i=1, i<=rows, i++ .

Step5:- Create nested for loop as j=1 ; j<=i ; ++j

Step6:- Display the no. as per user enters. the sequence from i

Step7:- Number increment to i++ from 1 .

Step8:- Display the space .

Conclusion:-

Thus we have successfully execute given expression on turbo C using nested for loop .

Nmodi

Scanned with CamScanner

Practical 5:-

Aim:- C program to find largest among numbers using array

Algorithms-

Step1:- Start Turbo C application.

Step2:- Declare the variable i and integer array a[10]

Step3:- Enter the for loop at i=0, i<10 and scans the value of a[i] till i<10. Exit the for loop.

Step4:- Enter the for loop at i=0, i<10. Use if conditional statement to check if a[0] < a[i] if true, put a[0]=a[i]

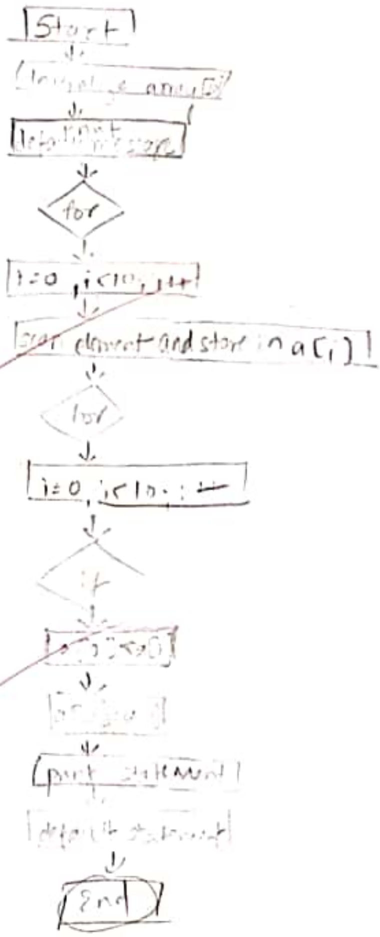
Step5:- Run the above for loop for i<10, Exit the ~~for~~ loop.

Step6:- Terminate the program.

Conclusion:-

Thus, we have executed the program successfully.

Flowchart -



Scanned with CamScanner

Output :-

```
Enter the elements
1

23

2
2
12
55

3

1
22
 100
The largest number is 100 .
```

Source Code :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
 int a[10],i;
 clrscr(),
 printf("Enter the elements of the list \n"),
 for(i=0;i<10;i++)
  {
   scanf("%d",&a[i]);
  }
 for(i=0; i<10; i++)
  {
     if (a[0]<a[i])
     {
       a[0]=a[i];
     }
  }
 printf("The largest number is %d", a[0]);
 getch();
}.
```

(5)

Aim:- C program to find even and odd numbers using array.

Algorithm:-

Step1- Start Turbo C application.

Step2:- Declare the variable i, num and array[100].
~~Print~~ Print the message for entering element.
Store it in num variable.

Step3:- Enter the for loop at i=0, i<num and increment it
store the element in array[i].

Step4:- Give the print message for even numbers.
Use for loop at i=0; i<num and increment it.
Use if condition for even numbers. Give the
print message ~~and~~ ~~st~~ for displaying.

Step4:- Use for loop at i=0, i<num, i++ and use if
condition to check the array element if odd

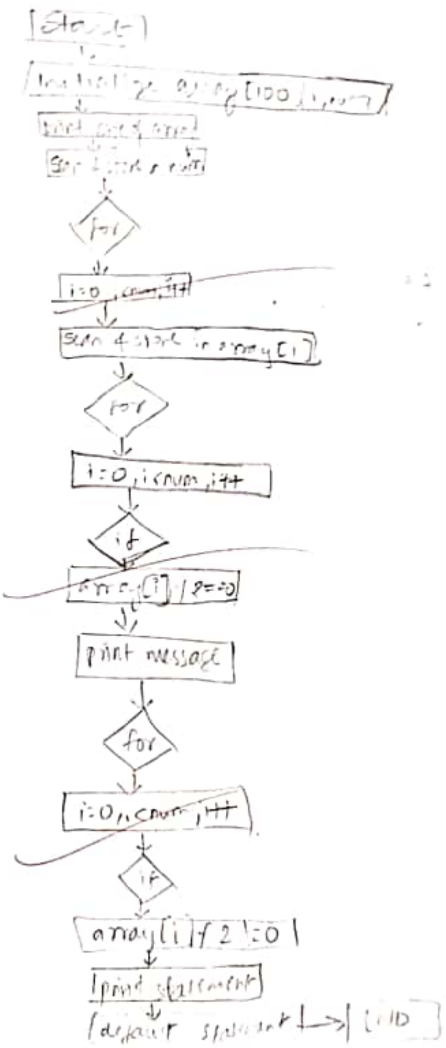
Step5:- Give the print message. Exit for loop.

Step6:- Terminate the program.

Conclusion:-

Thus, we have executed the program successfully

Flowchart -



Scanned with CamScanner

Output :-

```
Enter the size of the array :- 5
Enter the element of array :-
24
45
34
22
21
Even no. in the array are :
24  34  22  80
odd no. in the array are :
45  21
```

Source code :-

```
#include <stdio.h>
# include <conio.h>
{
int array[100], i, num;
printf("Enter the size of the array\n");
scanf("%d", &num);
printf("Enter the element of array\n");
for (i=0; i<num; i++)
{ {
   scanf("%d", &array[i]);
  }
  printf("Even no in the array are :");
  for (i=0; i<num; i++)
   {
     if (array[i] %2==0)
      {
        printf("%d\t", array[i]);
        }
        }
         printf("\n odd no. in the array ar
         for (i=0; i<num; i++)
          {
           if (array[i]%2 !=0)
           {
             printf("%d\t", array[i]);
             }
             }
              getch();
              }
```

Q

Aim:- C program to find average and sum using array.

Algorithm -

Step1 - start Turbo C application

Step2:- Declare the float int variable n, i. Initially num[100], sum=0.0, avg.

Step3:- ~~For loop~~ Using for loop at i=0; i<n; i++. Give print message and increment i by 1.

Step4:- Declare sum variable and store it by adding num[i].

Step5:- Average is sum divided by n.

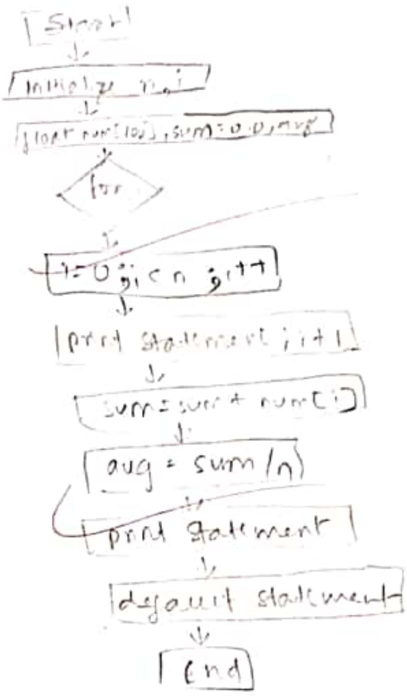
Step6:- Give print statement for average and sum.

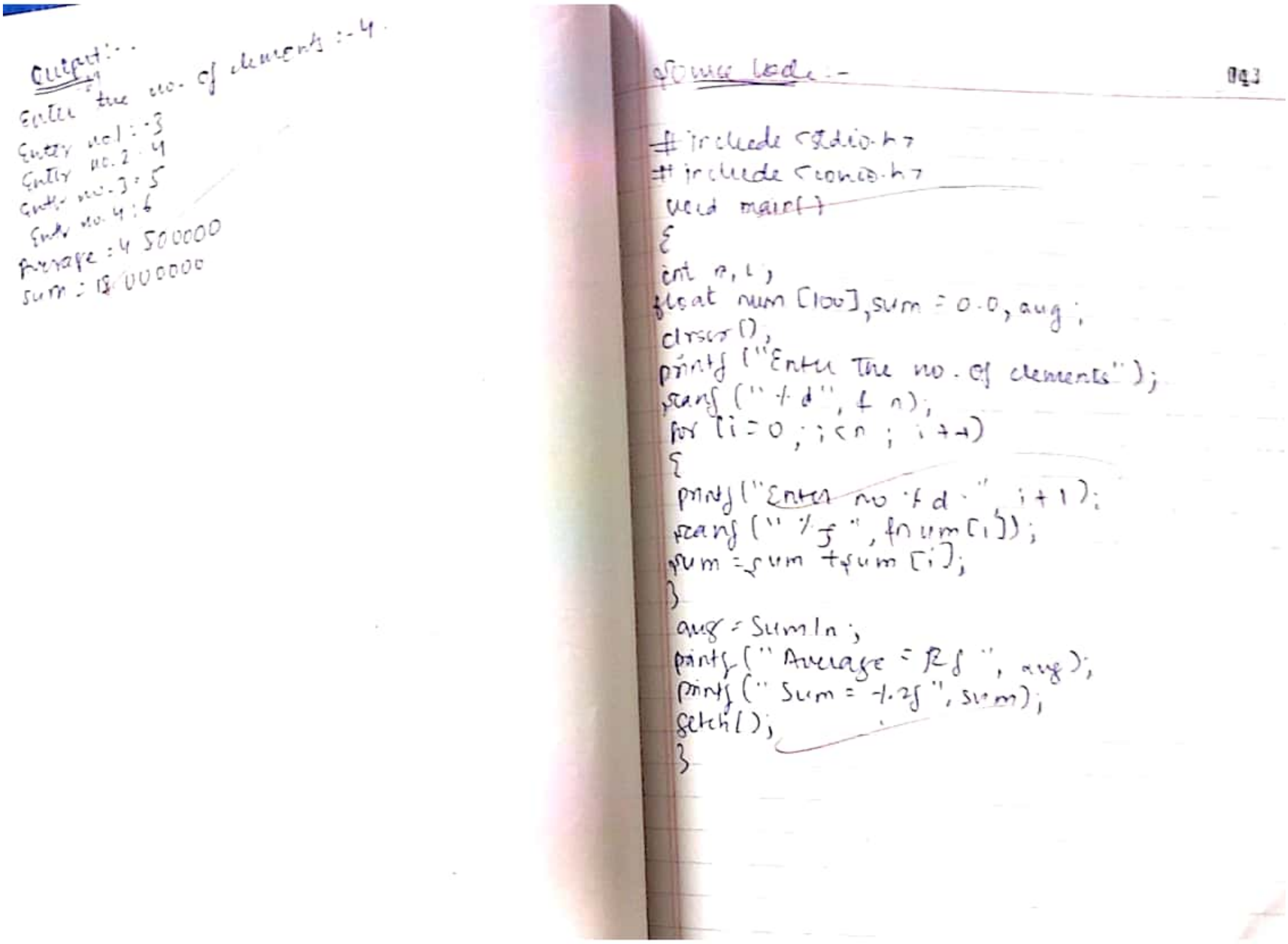
Step7:- Terminate the program.

Conclusion:-

Thus, we have executed the program successfully.

Flowchart -





Output:-

```
Enter the no. of elements :- 4.
Enter no.1 :- 3
Enter no.2 : 4
Enter no.3 : 5
Enter no.4 : 6
Average : 4.500000
Sum : 18.000000
```

Source Code:-

```
#include <stdio.h>
#include <conio.h>
void main()
{
int n, i;
float num[100], sum = 0.0, avg;
clrscr();
printf("Enter The no. of elements");
scanf("%d", &n);
for (i=0 ; i<n ; i++)
{
printf("Enter no %d : ", i+1);
scanf("%f", &num[i]);
sum = sum + num[i];
}
avg = sum/n;
printf("Average = %.2f", avg);
printf("Sum = %.2f", sum);
getch();
}
```

Practical 6

(7)

Aim:- To find factorial using recursion.

Step1:- Start Turbo C application.

Step2:- Declare the int variable factorial, n.

Step3:- Use if conditional statement and return factorial, and use else statement for returning 1.

Step4:- Declare int variables n, a.

Step5:- Use print statement for taking input from user.

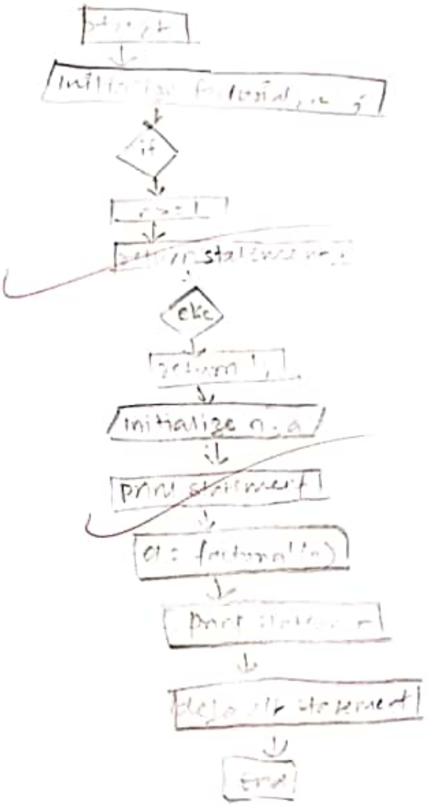
Step6:- Factorial of n is a.

Step7:- Use default statement.

Step8:- Display the output.

Step9:- Terminate the program.

Flowchart:-



Output:-

```
Enter a positive integer : 5
Factorial of 5 is 120.
```

Source Code :-

```
#include<stdio.h>
# include <conio.h>
int factorial (int n)
{
If (n>=1)
        return n * factorial (n-1);
else
        return 1;
}
void main()
{
int n, a;
printf (" enter a positive integer");
scanf (" %d", &n);
a= factorial (n);
printf (" \n factorial of %d is %d", n, a);
getch();
}.
```

(B) Aim:- C program which shows the use of get fu

Algorithm:-

Step1:- start Turbo C application.

Step2:- Declare character variable ch.

Step3:- Use print statement of entering any key to contin

Step4:- Use getch() function.

Step5:- Use print statement for entering alphabet.

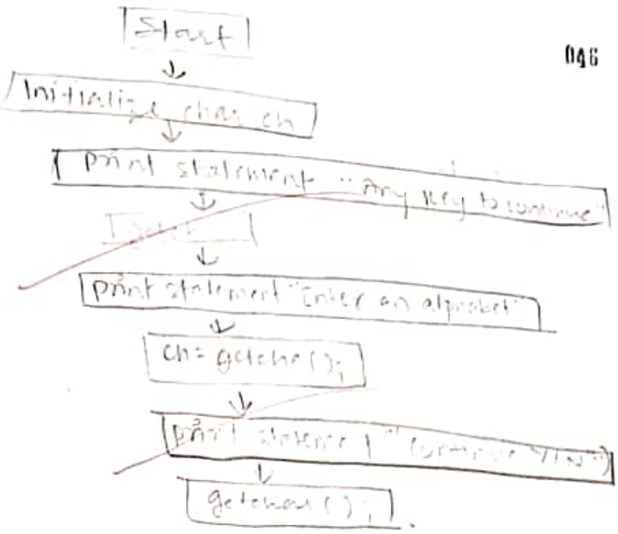
Step6:- Assign ch to getche() function.

Step7:- Use print statement for ~~conb~~ asking to continue the process.

Step8:- Use getchar() function.

Step9:- Terminate the program.

Flowchart:-


Start
Initialize char ch
print statement "Any key to continue"
getch
print statement "Enter an alphabet"
ch = getche();
print statement "Continue Y/N"
getchar();

Output :-

```
Press any key to continue
Enter an alphabet S
continue Y/N -N.
```

Source code :-

```
# include <stdio.h>
# include <conio.h>
void main()
{
char ch;
 printf ("\n Press any key to continue");
 getch();
printf ("\n Enter an alphabet");
 ch= getche ();
printf ("\n continue Y/N");
 getchar();
}.
```

(i) Aim:- C program for showing the use of put functions.

Algorithm:-

Step1:- Start Turbo C application.

Step2:- Initialize char ch as b.

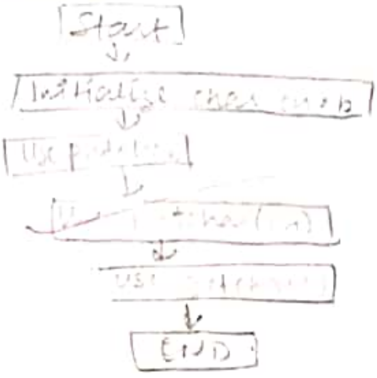
Step3:- Use putch function for ch.

Step4:- Use put char function.

Step5:- Use getchar function.

Step6:- Terminate the program.

Flowchart:-



Output:-

```
bb
```

Source Code:-

```
#include <stdio.h>
# include <conio.h>
void main()
{
 char ch='b' ;
 clrscr();
 putch(ch);
 putchar(ch);
 getchar();
}.
```

Conclusion:-

Thus we have executed the programs successfully.

Scanned with CamScanner

Practical · 7

① Swapping using Pointers :-

Aim :-

Algorithm :-

Step1 :- Start the Turbo C application.

Step2 :- Declare a fn prototype with two integer pointers as arguments before entering main().

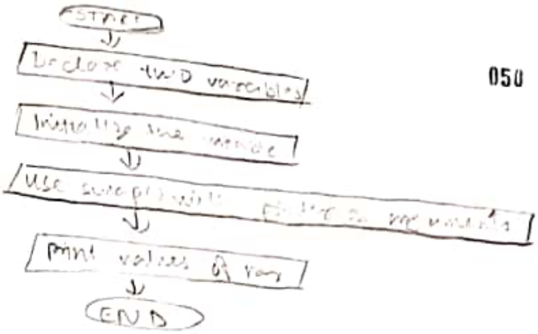
Step3 :- Declare the variable and accept their value from the user. Print the respective values using printf().

Step4 - Pass the address of the variables as arguments for the function.

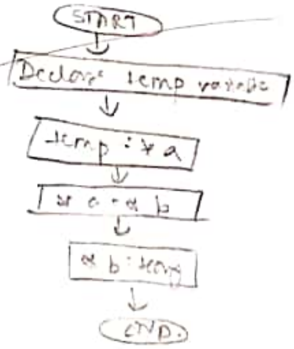
Step5 : Print the respective value of the variable.

Step6 :- Use the basic swapping algorithm in the function definition, but instead of normal variables, use pointers.

Flowchart.



Swap (*a ,*b):



Output -

```
[illegible]
2 4
[illegible] 12 [illegible] 24.
[illegible] 24 [illegible] 12.
[illegible]
```

Source code :-

```
void swap (int *m, int *n);
# include < conio.h>
# include < stdio.h>
void main ()
{
  int x, y;
  clscr();
  printf ("Enter the two numbers to be swapped");
  scanf ("%d %d, &x , &y);
  printf ("The values before swapping are %d and %d
           respectively", x, y);
  swap(&x, &y);
  printf (" The values after swapping are %d and %d
           respectively", x , y);
  getch();
}
void swap (int *m, int *n)
{
   temp = *m;
    *m = *n;
    *n = temp;
} -
```

7) SORTING OF ARRAYS USING POINTERS.

ALGORITHM:-

Step1:- Start Turbo C application.

Step2:- Initialise an integer array, i, j and temp values

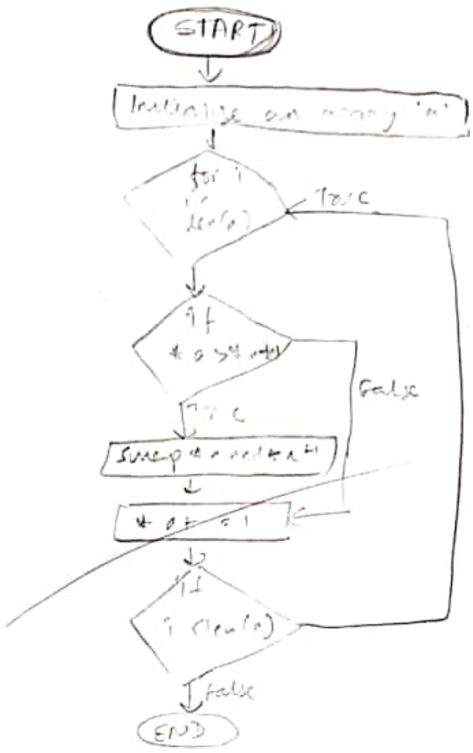
Step3:- Run a nested loop for i = 0 to len (a) and j = 0 to len (a)-1

Step4:- If *a > *a+1, swap the two values using basic swapping logic.

Step5:- Print the swapped array.

Step6:- Terminate the program.

Flowchart:-



Output:

```
Enter elements into the array:
1
6
7
8
2
9
10
11
5
10
{1,2,5,6,7,8,9,10,10,11} is the sorted array.
```

Source code -

```
# include <stdio.h>
# include <conio.h>
void main()
{
 int a[10]; i, j, temp;
 clrscr();
 for (i=0, i<10, i++)
 {
  for (j=0 ; j<10-i ; j++).
  {
      if (*a > *a+1)
      {
       temp = *a+1;
      *a+1 = *a;
        *a = temp;
      }
 } }
 printf ("%d is the sorted array", a);
 getch();
}.
```

(iii) "Traversal of one-directional array using pointers.

Aim:- Program on pointers.

Algorithm:-

Step1:- Start 'Turbo C' application.

Step2:- Declare an integer array and a variable i

Step3:- Run a while loop with 0 to length of array

Step4:- Print the default statements.

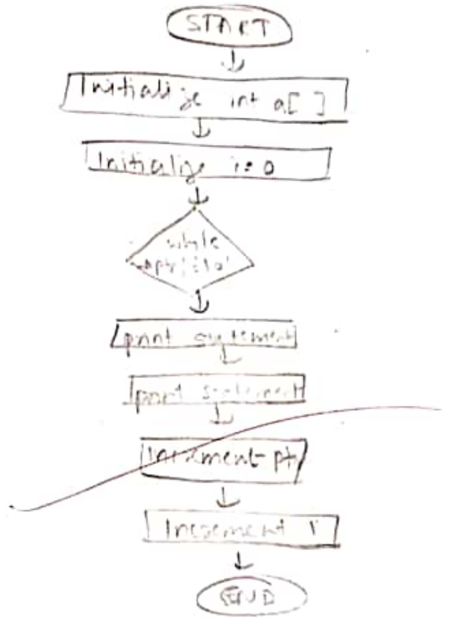
Step5:- Increment pointer and i.

Step6:- Terminate the program.

Conclusion:-

Thus we have executed programs based on pointers successfully.

Flowchart:-



Scanned with CamScanner

Output:-

```
The value of a [1] = 9
The address of a [2] = 65520
The value of a [2] = 4
The value of a [3] = 65522
The address of a [3] = 8
The value of a [4] = 65524
The address of a [4] = 2
The value of a [0] = 65516
The address of a [0] = 7
The value of a [0]
The address of a [1] = 65518
The value of a [1] = 9
The address of a [2] = 65520
The value of a [2] = 4
The address of a [3] = 65522
The value of a [3] = 8
The address of a [4] = 65524
The value of a [4] = 2.
```

Source code:-

```
# include < stdio.h>
# include < conio.h >
void main()
{
int a [5] = {7, 9, 4, 8, 2};
int * ptr ;
int i = 0;
ptr = & a[0];
while (*ptr != '\0')
{
printf ("\n\nThe address of a [%d] = %[illegible]", i, ptr);
printf ("\n The value of a[%d] = %d", i, *ptr);
ptr ++;
i ++ ;
}
getch();
}.
```

Scanned with CamScanner

Scanned with CamScanner

Practical: 8

A) Create a simple structure named student that holds the following variable : ① id ② CGPA ③ name .

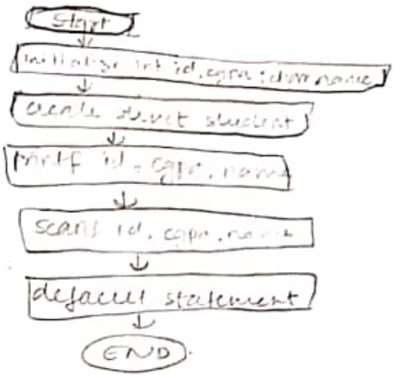
Algorithm:-

Step1:- Start Turbo 'C' application.
Step2:- Declare int 'id, cgpa and character name' of length 10.
Step3:- Create structure of name student.
Step4:- Create structure with pointer.
Step5:- Print statements for entering id, name and cgpa. using printf function.
Step6:- Store the information using scanf.
Step7:- print default statements.
Step8:- Terminate the program.

Conclusion:-

Thus, we have successfully executed the program.

Flowchart:-



Scanned with CamScanner

Output:-

```
Enter id 12
Enter cgpa 9
Enter name: rohan
* * * * * * *
Id = 12
cgpa = 9
Name = rohan
```

Source code:-

```
#Include <conio.h>
# include <stdio.h>
struct student
{
  int id, cgpa;
  char name [10];
};
void main()
{
    struct student * p;
    clrscr();
    printf ("\n Enter id:");
    scanf (" %d", &p->id);
    printf ("\n Enter cgpa :");
    scanf("%d", &p->cgpa);
    printf ("\n Enter name :");
    fflush (stdin);
    scanf (" %s", p->name);
    printf ("\n * * * * * * * * \n");
    printf ("\n Id = %d \n Cgpa = %d \n Name = %s", p->id,
                p->cgpa, p->name);
    getch();
}.
```

ii) Use of struct {'

Algorithm:-

Step1:- Open the Turbo 'c' application.

Step2:- Define a struct and take the input as done previously.

Step3:- Now define a fn to calculate total marks obtained out of 600 using CGPA.

Step4:- Print this along with the student data.

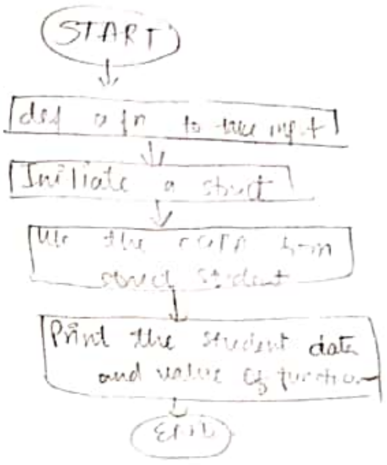
Step5:- Terminate the program.

Source code:-

```
def int calc (int n)
{
        return 60 * n
}

printf (" Total marks obtained " %d "; %d (        ")),
```

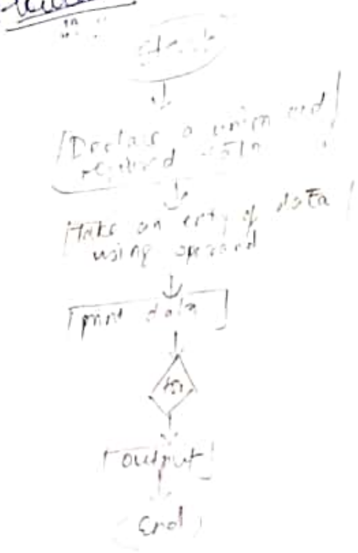
Flowchart:-


START
def a fn to take input
Initiate a struct
Use the CGPA from struct student
Print the student data and value of function
END

Output :-

```
Total marks    : 600
Total marks obtained : 540
```

Flowchart :-


Start
Declare a union and required data
Make an entry of data using operand
print data
i=n
output
End

ii) Union :-

Algorithm :-

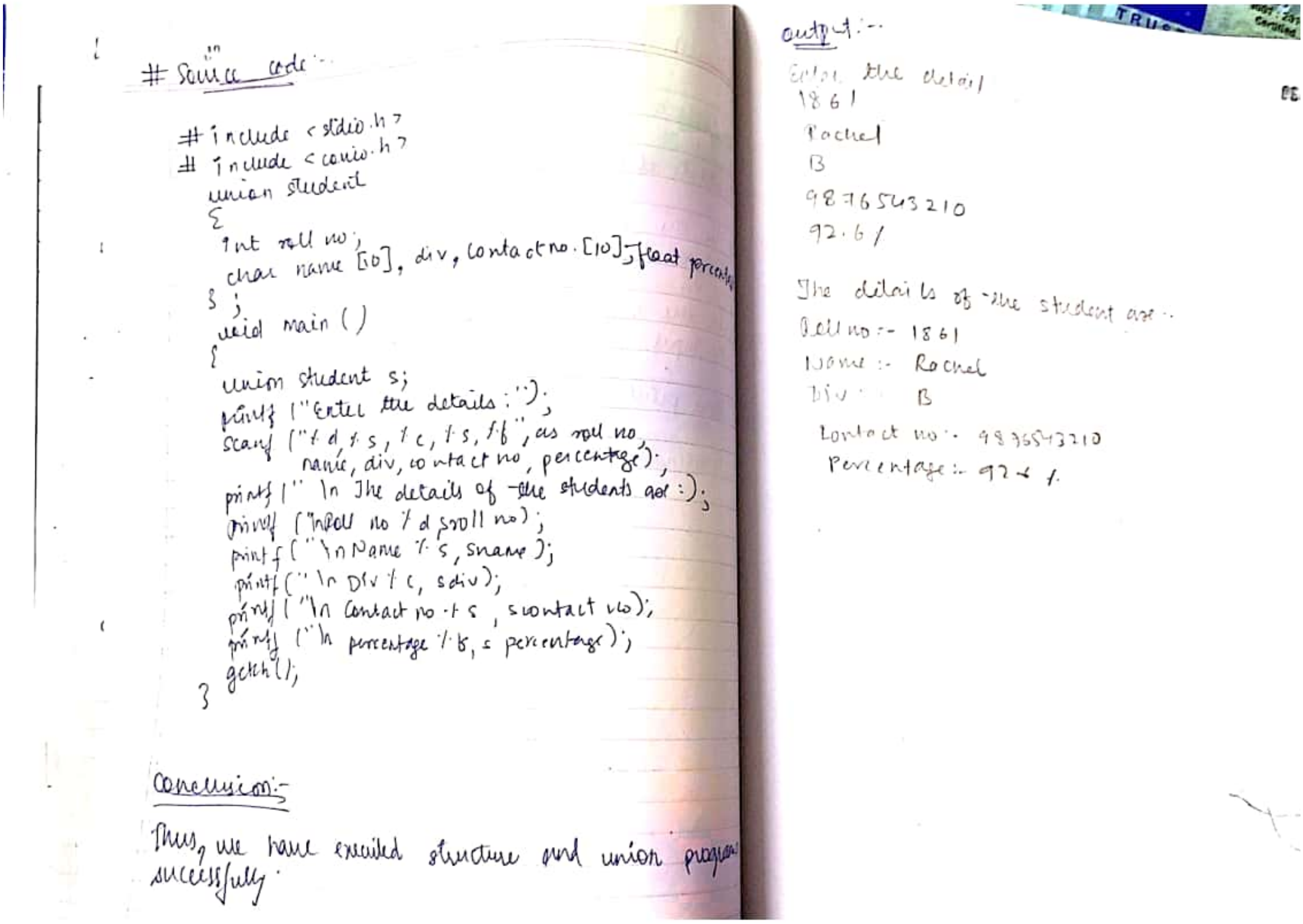
Step1:- Start the turbo c application.

Step2:- Use the union keyword to declare the union of different datatypes.

Step3:- In the main body of the function, use the '.' to take the input.

Step4:- Now print all the data of union.

Step5:- Terminate the program.



Source code :-

```
#include <stdio.h>
# include <conio.h>
  union student
  {
   int roll no;
   char name [10], div, contactno.[10];float preceta
  };
  void main ()
  {
   union student s;
   printf ("Enter the details :");
   scanf ("%d %s, %c, %s, %f", as roll no,
          name, div, contact no, percentage);,
   printf (" \n The details of the students are :);
   printf ("\nRoll no %d sroll no);
   printf (" \n Name %s, sname );
   printf (" \n Div %c, sdiv);
   printf (" \n Contact no %s , scontact no);
   printf (" \n percentage %f, s percentage);
   getch();
  }
```

Conclusion:-

Thus, we have executed structure and union program successfully.

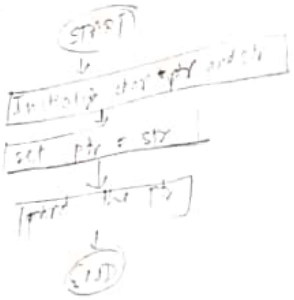
output :-

```
Enter the detail
1861
Rachel
B
9876543210
92.6%

The details of the student are :-
Roll no :- 1861
Name :- Rachel
Div :  B
 Contact no :- 9876543210
 Percentage :- 92.6 %
```

Scanned with CamScanner

Flowchart:-



Output:-

```
Enter a string : C Programming language.
The copied string is : C Programming language.
```

Practical-9

Aim:- Various operations on strings in 'c'.

1) Copying a string (without in-built function).

Algorithm:-

Step1- Open the Turbo 'c' application.
Step2- Declare a pointer to a char and character array (string)
Step3- Get the value of the string using gets().
Step4- Let pointer equal to character array.
Step5- Print the pointer variable's value
Step6- End the program.

Source code:-

```
#include <conio.h>
# include < stdio.h >
void main ( )
{
    char *t, s[25] ;
    clrscr( );
    printf (" Enter a string :");
    gets( s) ;
    t = s;
    printf (" \n the copied string is %s ", *t));
    getch();
}
```

ii) String library functions:-

strcpy (destination, source): used to copy one string into anot

strlen (): used to ~~st~~ find the length of a string.

strcat (): used for concatenation of 2 strings.

Algorithms:-

Step1:- Start the Turbo 'C' application.

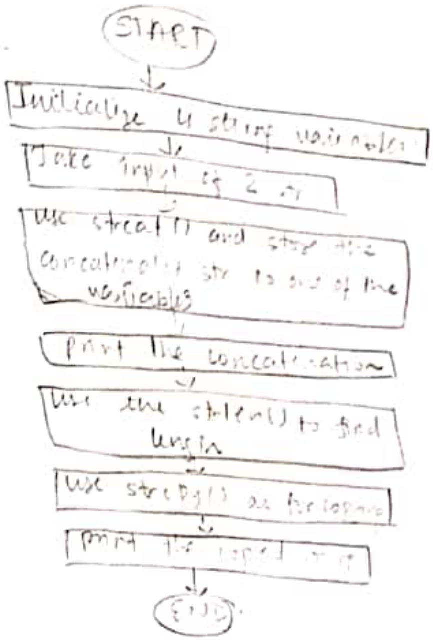
Step2:- Declare a few string variable.

Step3:- Take the input for two of the variables.

Step4:- Use the above mentioned functions and point the results.

Step5:- Terminate the program.

Flowchart :-


START
Initialize 4 string variables
Take input of 2 str
Use strcat() and store the concatenating str to one of the variables
Print the concatenation
Use the strlen() to find length
Use strcpy() as for copying
Print the copied str
END

Output:-

```
Enter any 2 strings
Rachel
Green
 The concatenated string is: Rachel Green
 The length of string is 11
 Copy of the original string : Rachel
```

Source code :-

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
void main()
{
 char a[20], b[20], c[20], d[20];
 clrscr();
 printf("Enter two strings : \n");
 gets(a);
 gets(b);
  c= strcat(a, b);
  printf(" The concatenated string is: %s", c);
  printf(" The length of string : %d", strlen(c));
  strcpy(d, a);
  printf(" copy of the original string: %s", d);
  getch();
}
```

iii) Length of string (without string library).

Algorithm:-

Step1:- Start the Turbo 'C' application

Step2:- Initialize a character array and pointer. Also initialize an ~~point~~ integer variable to zero.

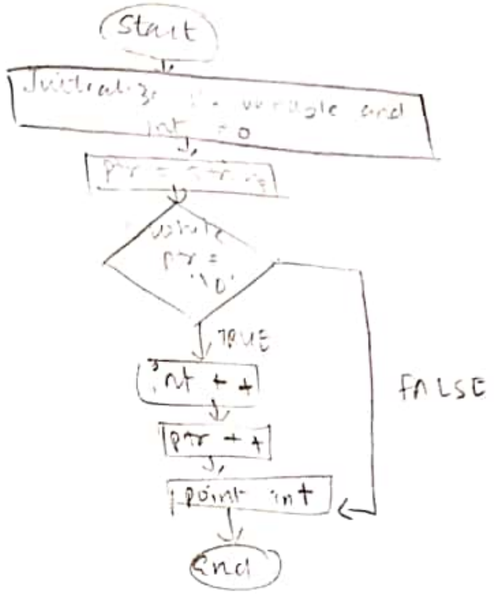
Step3:- Put the pointer equal to the string.

Step4:- Using a while loop, increment the integer variable while the pointer is not null ('\0'). Also, increment the pointer variable.

Step5:- Print value of integer.

Step6:- Terminate the program.

Flowchart :-



Output:-

```
Enter a string: This is a text
length of string: 14.
```

Source code:-

```
#include <conio.h>
#include <stdio.h>
void main()
{
  char *t, a[25];
  int flag = 0;
  printf("Enter a string: ");
  gets(a);
   t = a;
   while(*t != '\0')
   {
       flag++;
       t++;
   }
   printf("\n length of string: %d", flag);
    getch();
}
```

Conclusion:-

Thus we have executed programs based on strings successfully.