```
#include <stdio.h>
#include <conio.h>
void main()
{
    int roll_no;
    char name[20], mobile_no[];
    float percentage;
    clrscr();
    printf("enter students name\n:");
    scanf("%d", &name);
    printf("enter student's roll no\n:");
    scanf("%d", &roll no);
    printf("enter student's mobile no\n:");
    scanf("%s", &mobile no");
    printf("enter students 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:
Nidhi
Enter student's roll no:
1876
Enter student's percentage:
90.20
Enter student's mobile no:
0111101111
```

```
student's name: Nidhi
student's roll no: 1876
student's percentage: 90.20
students mobile no: 0111101111
```


Practical - 1

Aim: Write a C program to understand basic datatypes & I/O

Theory: Writing a program to display student's name, roll_no, mobile no & percentage

Algorithm:

Step 1 - Declare a variable name roll no as integer, also declare name, mobile no as characters & percentage as float.

Step 2 - use printf function to print questions for user in order to give input.

Step 3 - use scanf function to read user's input & store in its allocated memory.

Step 4 - Again use printf function to display the output.

Conclusion: The given program gives user an idea about how to build in datatypes work in C & also about how user can give input & display output.

Nidhi

PSO Practical - 02

Aim: Write a program on operation of expressions.

Theory:

Write a program to create a dynamic calculator.

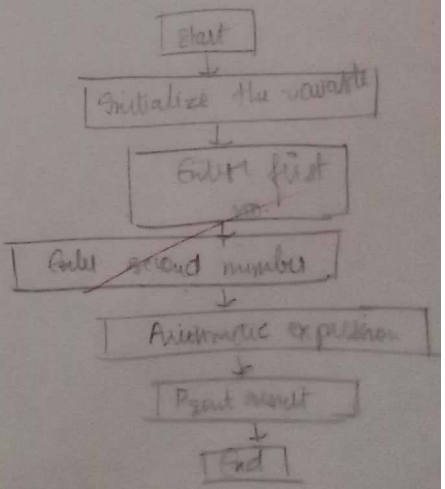
Algorithm:

Step 1- Declare a variable name for first & second number of integer.
Step 2- Now use scanf function to receive input from user
Step 3- Now to add two numbers given by user, use the expression num1 + num2.
Step 4- Now to subtract two numbers given by use, use expression num1 - num2
Step 5- Again use expression num1 * num2 if user wishes to multiply the two inputs.
Step 6- Use expression num1 / num2 if user wishes to divide the two inputs.
Step 7- Now use printf function to display output.


```
#include <stdio.h>
#include <conio.h>
void main();
{
    int x, y, z;
    clrscr();
    printf("Enter two numbers:\n");
    scanf("%d %d", &x, &y);
    z = x + y;
    printf("sum of two numbers = %d\n", z);
    scanf("%d %d", &
    z = x - y;
    printf("substraction of two numbers = %d\n", z);
    z = x * y;
    printf("multiplication of two numbers = %d\n", z);
    z = x / y;
    printf("Division of two numbers = %d\n", z);
    z = x % y;
    printf("Modulus division of two numbers = %d\n", z);
    getch();
}
```

output -

Enter two numbers: 2
Enter two numbers: 2
sum of two numbers = 4
subtraction of two numbers = 0
division of two numbers = 1



PRACTICAL -2

b] write a program in C to explain ternary operator

step 1 - Declare variables a, b & x as integers.

step 2 - store the value of a as 5 & store the value of b as 15.

step 3 - Now to compare between who is greater use ternary operator x to find.

step 4 - Use printf function to display output.

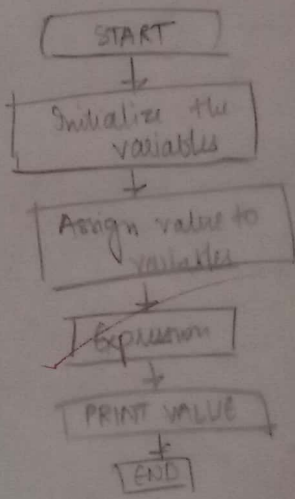
Conclusion : These programs help us & in having better understanding about operator & expressions.

b]

```
// 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

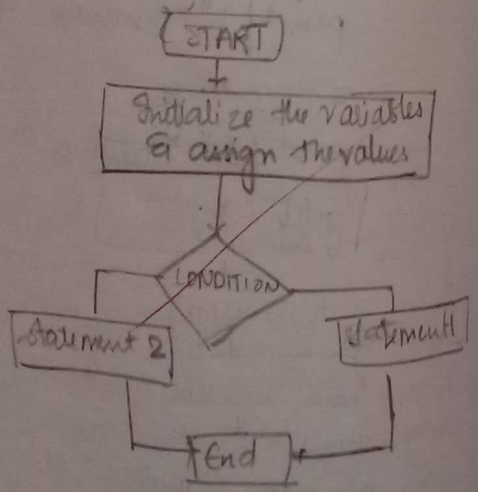


if -

```
#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



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

Theory :

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

step 1 - Declare a variance as integer & assign its value ie 20

step 2- Now do compare whether 20 is greater than 15 use if statement.

step 3- If the condition is true, printf that 20 is less than 15 & if condition is false skip the if statement & print I am not in if

b] write a program in C to explain if else statement.

Step 1 - Declare a variable as integer & assign its value i.e 20

Step 2 - Now to compare the given value if its greater or not we if else conditional statement.

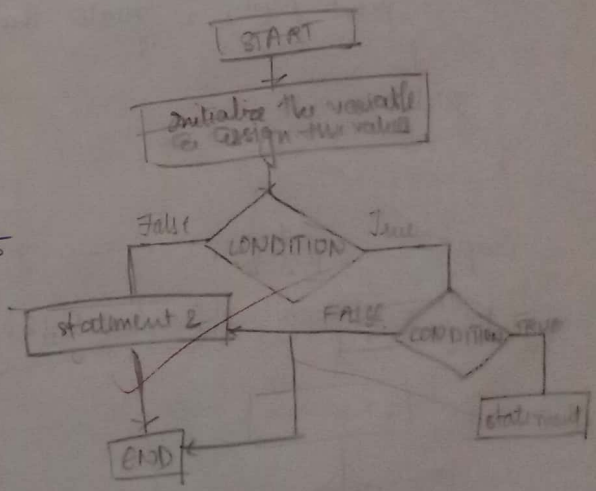
Step 3 - If condition is true the print 20 is less than 15 or if condition is false then print 20 is greater than 15

if else

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int i = 20;
  clrscr();
  if(i<15)
  {
    printf("i is smaller than 15\n");
  }
    else
     {
      printf(" i is greaterthan 15 \n");
     }
  getch();
}
```

Output -

20 is greater than 15

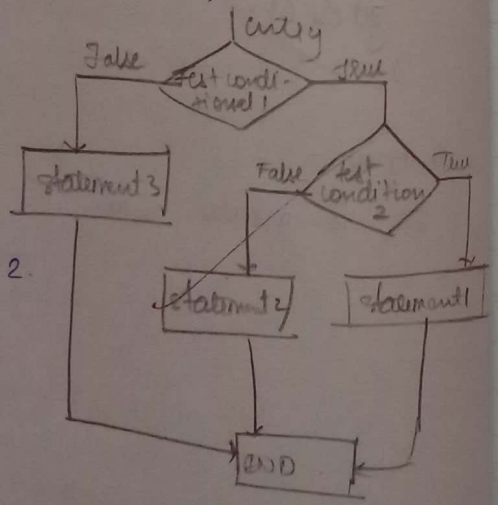


nested if

```
#include <stdio.h>
#include<conio.h>
void main()
{
    int i=10;
    clrscr();
    if(i==10)
    {
     if(i<15)
     printf("i is smaller than 15\n");
     if(i<12)
     printf("i is smaller than 12\n");
     else
     printf("i is greater than 15\n");
    }
  getch();
}
```

Output:

10 is greater than 10 & 12.



c] Write a program in C to explain nested if statement.

Algorithm:

step 1:- Declare a variable as integer and assign value to it.

step 2- Now use nested if logic to compare if given number is greater or not

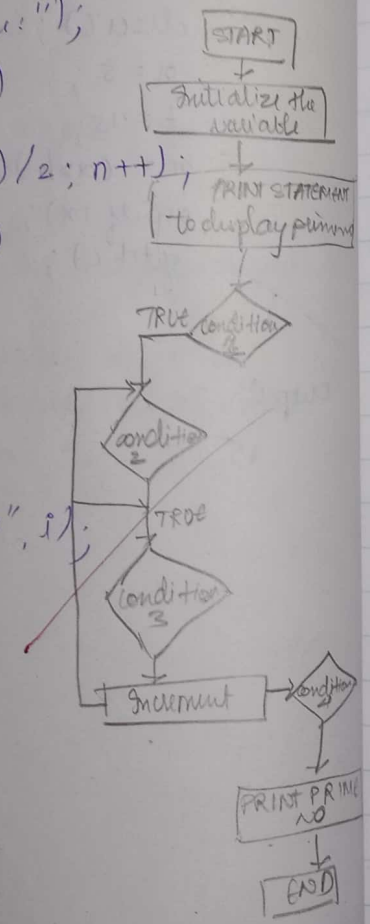
step 3- If first condition is true then go to second condition if second condition is also true then print that 20 is greater than 15 & 12. If one of the condition are not true then skip the part & print 20 is greater than 15 & 12.

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

Ahmad

PROGRAM :

```
#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+1)/2; n++);
        {
            if(i%n==0)
            {
                a++;
            }
        }
        if(a==0)
        {
            printf("%d\n", i);
        }
    }
    getch();
}
```



Output : The prime numbers are : 2
3
5
7
11
13
17
19

PRACTICAL - 4

Aim : a) To display prime numbers using for loop.

Algorithm :

step 1- Initialize three variables out of which two are loop variables & are in a count ~~test~~ variable

step 2- Initialize a for loop from 2 to 50. Set the count variable to zero

step 3- ~~Next~~ Put another loop within a loop in step 2 that goes from 2 to the first loop variable / 2.

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

step 5- Come out of the second loop & check whether the count variable is 0 if true print the number.

step 6- Terminate the program

Conclusion : Prime numbers were displayed using for loop.

b) Aim: Write a program on fibonacci series -

Algorithm :

Step 1 - Start the turbo c program.

Step 2 - Declare the variable, $n_1=0$, $n_2=1$ & number = 0.

Step 3 - Initialize the variable, $n_1=0$, $n_2=1$ & number = 0.

Step 4 - Enter the no. of terms fibonacci series to be printed.

Step 5 - Print first two terms of series as $n_1=0$ & $n_2=1$.

Step 6 - Use the for loop as per following step -

$n_3 = n_1 + n_2$

$n_1 = n_2$

$n_2 = n_3$

Increase the value of i element each time by 1.

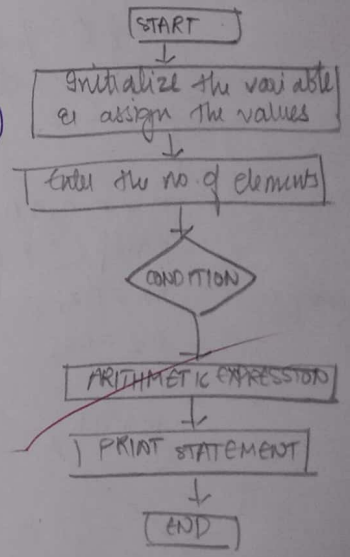
Step 7 - Print the value of the number.

Step 8 - Enter the program.

Conclusion - Thus we successfully executed fibonacci series of Turbo C.

~~OUTPUT~~ — PROGRAM —

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int n1=0, n2=1, n3, i, number;
  clrscr();
  printf("Enter no. of elements\n");
  scanf("%d", &number);
  scanf("%d", &n1, &n2);
  for (i=2; i<=number; ++i)
    {
      n3 = n1 + n2;
      printf("%d", &n3);
      n1 = n2;
      n2 = n3;
    }
  getch();
}
```



output -

Enter no. of elements : 6

0 1 1 2 3 5 8

PROGRAM -

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n1=0, n2=1, n3, i, number;
    printf("enter no. 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();
}
```

```
{ int n1=0, n, i, j;
  clrscr();
  printf("enter the no of rows");
  scanf("%d", &n);
  printf("\n");
  for(i=0; i<=n; j++)
  { for(j=0; j<=i; j++)
    { n1++;
      printf("%d", &n1);
    }
    printf("%c\n");
  }
  getch();
}
```

Output -

```
Enter the number of rows: 3
1
2 3
4 5 6
7 8 9 10
```


Q) Ans: Write a c program on following expression -

Step 1- Start the turbo c program.

Step 2- Initialize the program, rows, i, j, number = 1.

Step 3- Display the number of rows.

Step 4- Enter the for loop (i = 1, i <= rows, i++)

Step 5- Create nested for loop, 5 j = 1, j <= i, ++j).

Step 6 - Display the number as per uses enter from i=

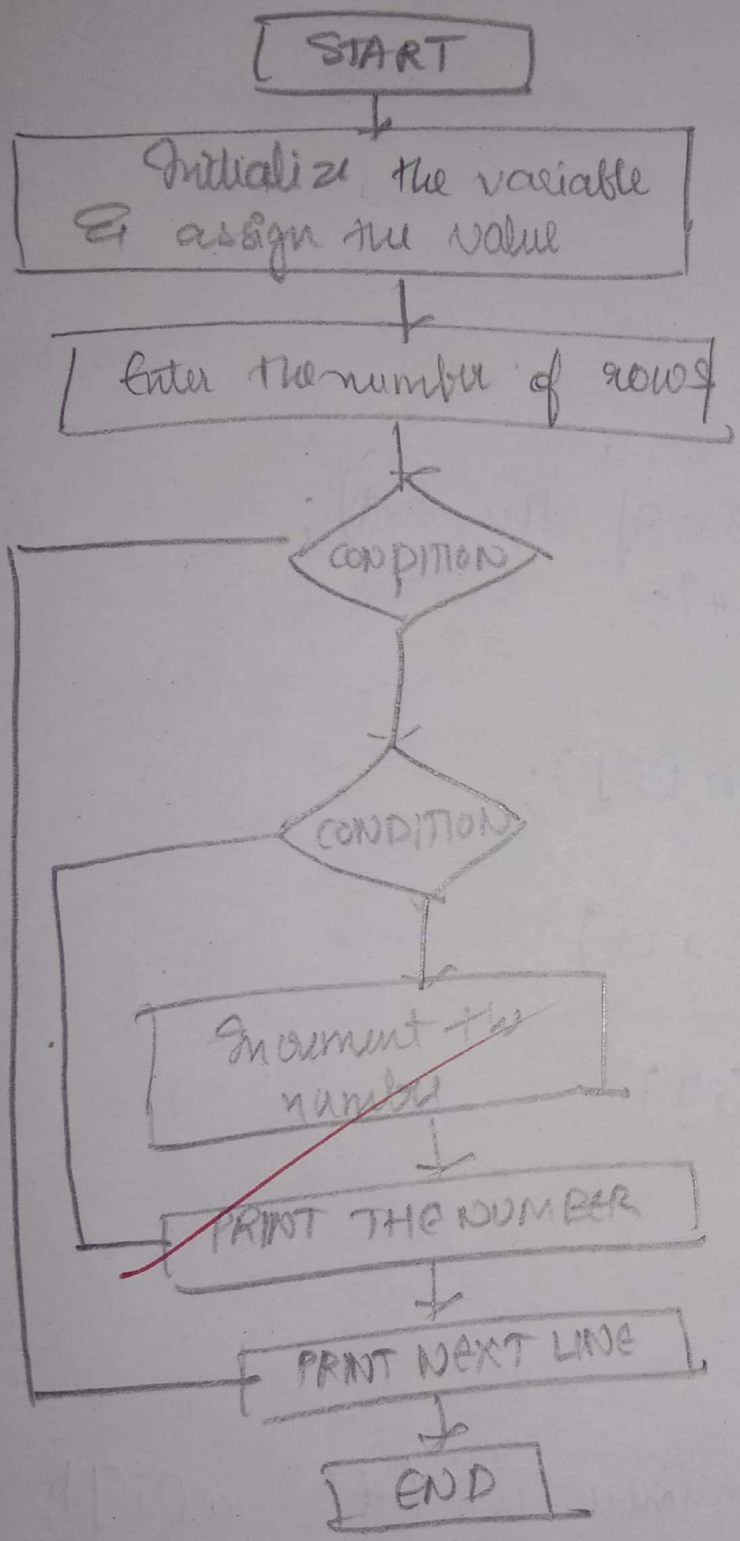
Step 7- Increment the numbu to 1.

Step 8 - Display the space.

Conclusion : Thus, we have successfully executed given expression on Turbo C using nested for loop.

Namoodi

FLOWCHART —



```
#include <conio.h>
#include <stdio.h>
void main()
{
    int a[10], i;
    clrscr();
    printf("Enter the element of the int");
    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[i]);
    getch();
}
```

output -

```
12    2    1
23    12   22
2     55   100
      3
```

The largest number is 100

Aim : A) Write a C program to find largest number among array.

Algorithm :

Step 1 - Start Turbo C applications.

Step 2 - Declare the variable i & integer array a[10]

Step 3 - Enter the for loop at i=0, i<10 & use the value of a[i] till i<10. Exit the for loop.

Step 4 - Enter the for loop at i = 0, i < 10 use if condition statement to check if a[0] < a[i] if true. put a[0] = a[i]

Step 5 - Run the above for loop for i < 10, exit the loop.

Step 6 - Terminate the program.

8) Aim : Write a C program to print the number of odd & even number in an array.

Algorithm :

Step1- Create an array. Take its size from user & define it's element using loop.

Step2- Display the size of array from user.

Step3- Display the element of array entered by user.

Step4- Take the initiator in a for loop using which all the elements of the array exist.

Step5- Display even number in the array from for loop.
if (array[i] % 2 == 0)
Display the even number from the given array.

Step6- Display the odd number
if (array[i] % 2 != 0)
then display the odd number from the array.

Step7- Terminate the Turbo C.

~~Out~~ CODE -

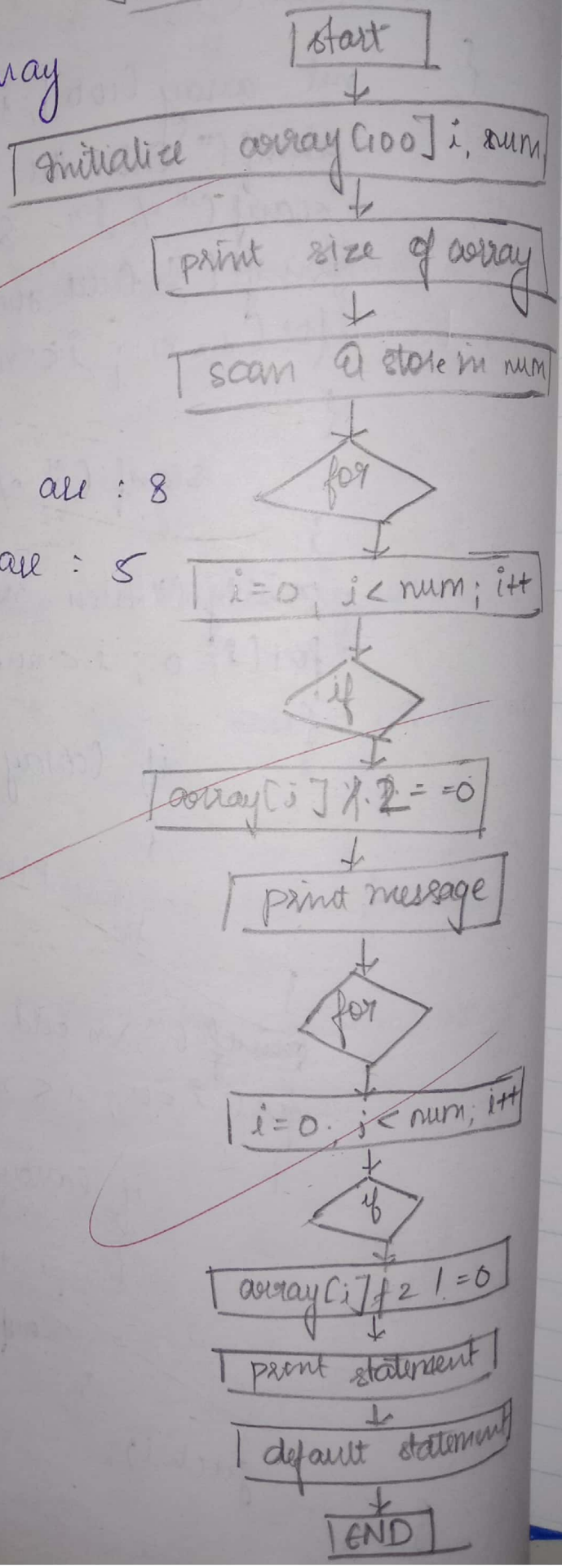
```
#include <stdio.h>
#include <conio.h>
void main()
{   int array[100], i, num;
    printf("Enter the size of the array\n");
    scanf("%d", &num);
    printf("Enter the element of array\n");
    for(i=0; i<num; i++)
    {
        scanf("%d", &array[i]);
    }
    printf("Given number in the array are:");
    for(i=0; i<num; i++)
    {
        if (array[i]%2==0)
        {
            printf("%d\t", array[i]);
        }
    }
    printf("\n odd numbers in the array are:");
    for(i=0; i<num; i++)
    {
        if (array[i] % 2 != 0)
        {   printf("%d", array[i]);
            scanf("%d", &num);
        }
    getch();
}
```

000

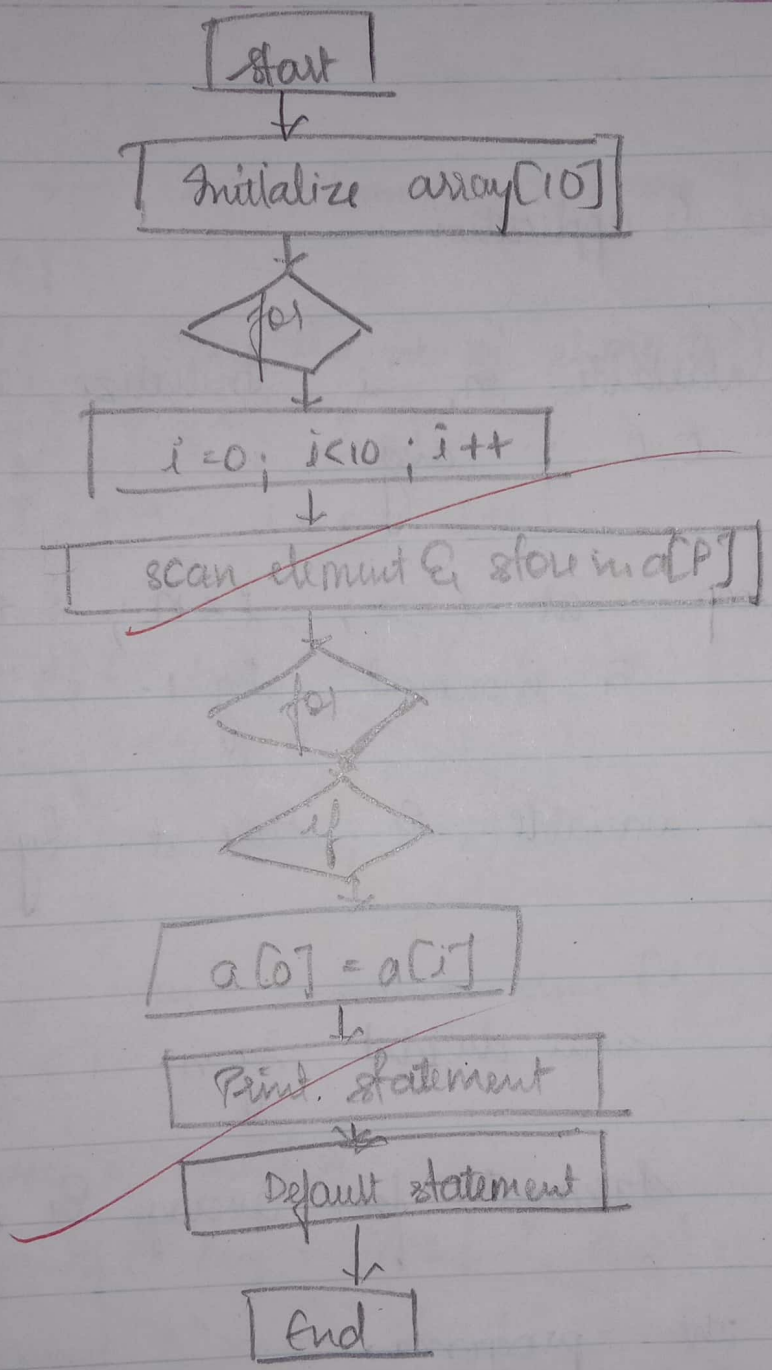
output -

Enter the size of the array
4
Enter the element of array
5
3
8
1
Even number in the array are : 8
Odd number in the array are : 5

Flowchart :-



Flowchart for largest number -



3) Aim : WAP to find the sum or average of element in an array

Algorithm -

Step 1 - start Turbo C application.

Step 2 - Declare int variable n, i. Initialize num[100], sum = 0.0 ; avg

Step 3 - Using for loop at i=0 ; i<n ; i++. give print message & increment by 1.

Step 4 - Declare sum variable & store it by adding num[i]

Step 5 - Average is sum divided by n.

Step 6 - Give print statement for average & sum.

Step 7 - Terminate the program.

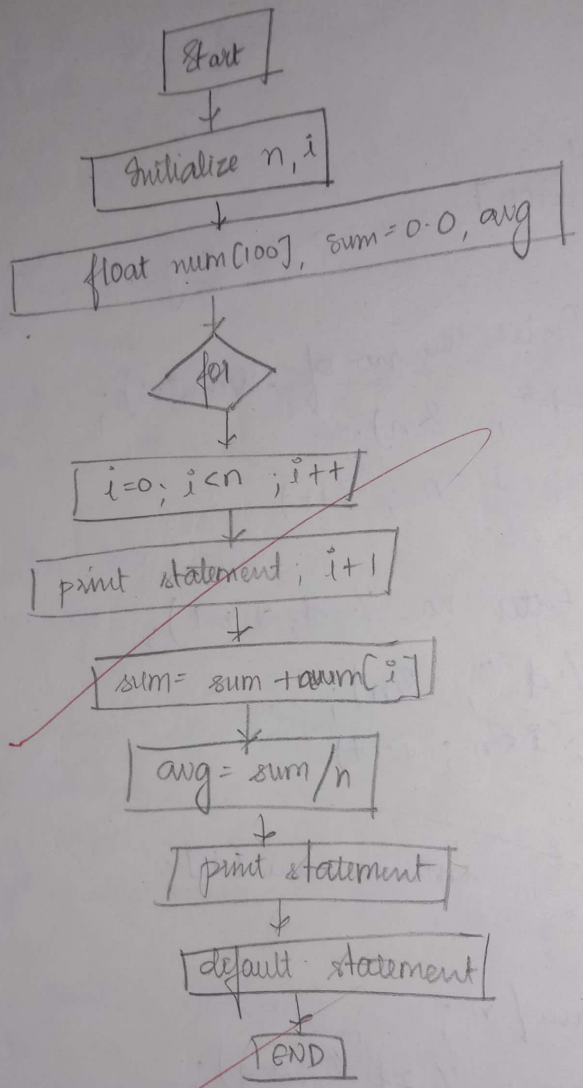
CONCLUSION - Thus, we have executed the program successfully.

Nhmodk

SOPE -


```
#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("%d", &n);
        for(i=0; i<n; i++)
        {
            sum = sum = sum[i];
        }
        avg = sum/n;
        printf("Avg = %2f", avg);
        printf("sum= %2f", sum);
        getch();
}
```

FLOWCHART -



OUTPUT - Enter the no. of elements : 4
Enter the no : 3
Enter the no : 4
Enter the no : 5
Enter the no : 6
Average : 5.00000
Sum : 15.00000

SAD PRACTICAL - 6

Aim - Factorial of a number using recursion.

Algorithm -

Step 1 - Start Turbo C application.

Step 2 - Declare integer factorial (int n)

Step 3 - Using if statement, check if n>=1 & return n* factorial (n-1) & else statement, return 1.

Step 4 - Initialize integer n, a & print statement by entering a positive integer.

Step 5 - Declare a = factorial (n)

Step 6 - Use print statement "factorial of %d is %d".

Step 7 - Terminate the program.

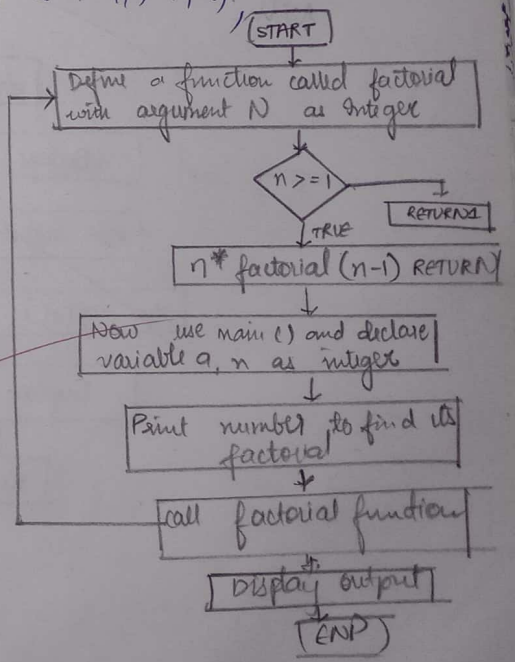
Conclusion: Thus, we have executed the program successfully


```
#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;
    clrscr();
    printf("enter a positive integer");
    scanf("%d",&n);
    a=factorial(n);
    printf("\n Factorial of %d is %d",n,a);
    getch();
}
```

OUTPUT -

Enter a positive integer: 6

Factorial of 6 is 720

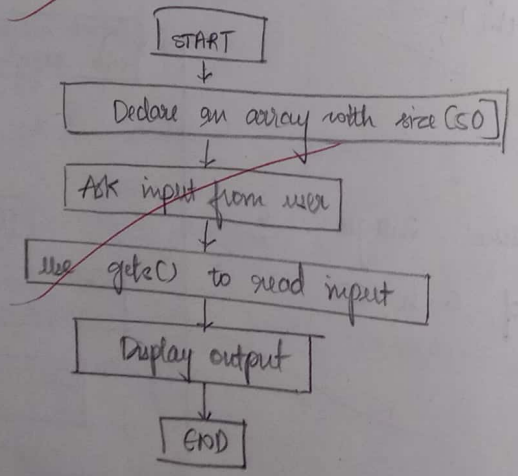


044

```
#include <string.h>
#include <stdio.h>
#include <conio.h>
void main()
{       char name[50];
        printf("\n Please enter your name :");
        gets(name);
        printf("\n\s", name);
        getch();
}
```

OUTPUT -

Please enter your name Vidhi
Vidhi



2) Aim : WAP to which shows the use of gets() function.

Algorithm -

Step 1 - Open Turbo C application.

Step 2 - Initialize character name [50].

Step 3 - Use print statement to display your name.

Step 4 - Use gets(name) to display read the input from user.

Step 5 - Print your name on screen

Step 6 - Terminate the program.

Conclusion : Thus, we have executed the program successfully.

3) Aim - WAP to show the use of puts() function.

Algorithm -

Step 1 - Declare a variable name as array with size 50 with character datatype.

Step 2 - Now ask user to enter your name.

Step 3 - Now use puts function to display your output

Step 4 - Terminate the program.

Conclusion: Thus, we, have executed the program successfully

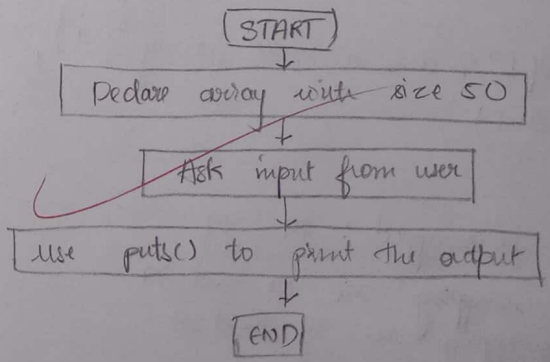
Manali


```
#include <string.h>
#include<stdio.h>
#include<conio.h>
void main()
{
    char subject[60];
    clrscr();
    printf("enter a subject:");
    gets(subject);
    printf("\n entered subject is:");
    puts(subject);
    getch();
}
```

OUTPUT -

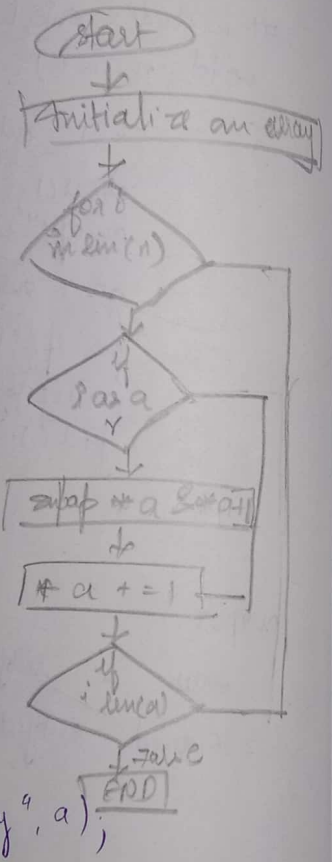
Enter a subject: maths

Entered subject is maths



#CODE (sorting)

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



#Output-

Enter number to be swapped : 29 99

The no before swapping are 29 and 99

The no after swapping are 99 and 29

PRACTICAL - 7

Aim : Swapping of array

Algorithm :

Step 1 : Start the Turbo C program.

Step 2 - Declare a function prototype with two integer pointer as argument before entering main().

Step 3 - Declare 2 variable & accept the value from the user. Print the respective value using printf()

Step 4 - Pass the address of the application as argument for the function.

Step 5 - Print the respective value of variable.

Step 6 - Use the basic algorithm in the function definition instead of normal variable user.

Conclusion - We have successfully executed program.

ii) Sorting of array using pointer

Algorithm:

Step 1 - Initialize an integer array & temp variable

Step 2 - Run an nested loop as i=0 len(a) and OR i=0 to len(a) -i

Step 3 - y+a > *a+1, swap the two values using basic swapping logic

Step 4 - Print the swapped array

Step 5 - Terminate the program.

Conclusion - We have successfully executed program.

#OUTPUT :-

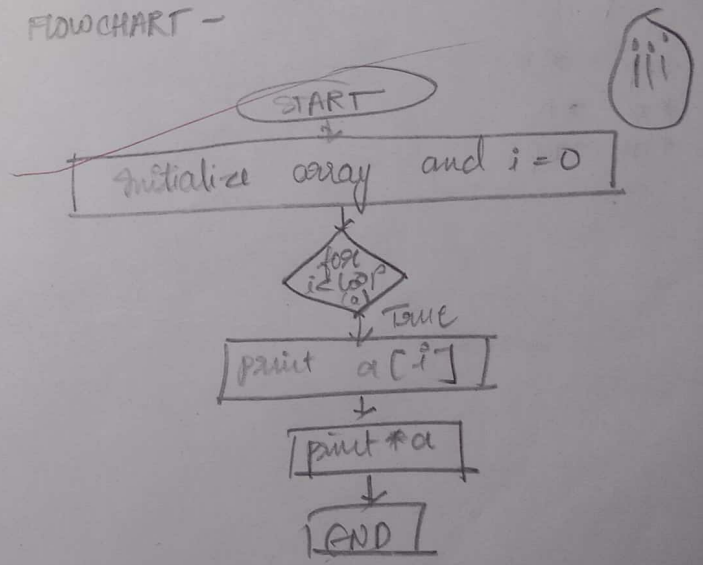
Insert element into the array

1
9
8
8
2
3
6
11

{1,2,3,5,6,8,9,11}

is the sorted array

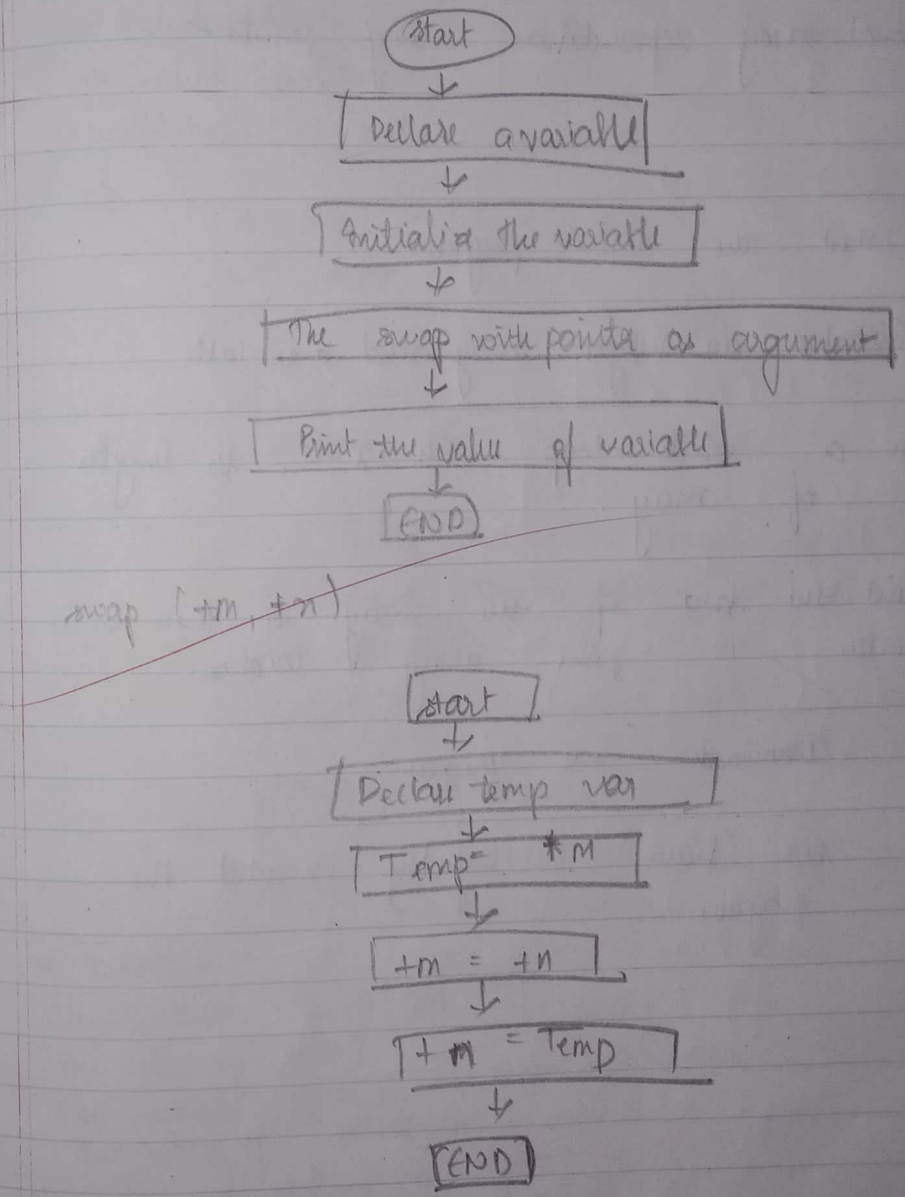
FLOWCHART -



#code

~~void sort (int x, int *p)~~

```
#include <stdio.h>
#include <conio.h>
void swap (int* m, int *n);
void main()
{
    ~~int a[10], i, temp;~~
    int x, y;
    clrscr();
    printf("Enter the two number to be swapped: ");
    scanf("%d %d", &x, &m)
    printf(" The value of before swapping are %d and %d
            respectively x, y);
    getch();
}
void swap (int *m, int *n).
{
    temp = *m;
    *m = *n;
    *n = temp;
}
```

iii) One dimensional array representation using pointer -

Algorithm :

Step 1 - Start the Turbo C program.

Step 2 - Initialize an integer array and a variable

Step 3 - Run a while loop with i=0, to length of array.

Step 4 - Print the data of the array & then use pointer to print array location.

Step 5 - Terminate the program.

Conclusion - We have successfully executed the program.

Unsolved

#CODE -

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

OUTPUT -

The address of a[0] = 65516
The value of u[0]
The address of a[1] = 65518
The value of u[1]
The address of a[2] = 65520
The value of u[2]
The address of a[3] = 65522
The value of u[3]
The address of a[4] = 65524
The value of u[4]

CODE:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    struct student
    {
        int id;
        float CGPA;
        char name[10];
    };
    struct student s[20];
    int size, i;
    clrscr();
    printf("\n Howmany words you want to insert:");
    scanf("%d", &size);
    for(i=1; i<size; i++)
    {
        printf("\n %d Enter id:", i);
        scanf("%d", e[i].id);
        printf("\n %f Enter CGPA:", i);
        scanf("%f", e[i].CGPA);
        printf("\n %C Enter name:", i);
        scanf("%c", e[i].name);
    }
    printf("\n\n Students Data \n\n");
    printf("\t id \t CGPA \t name");
    for(i=1; i<=size; i++)
```


PRACTICAL - 8

A] Create a simple structure named student that holds the following variable
i) id ii) CGPA iii) Name

Algorithm:

Step 1 - Open turbo C++ program. As per question use struct student.

Step 2 - Initialize int for id, float for CGPA & char for Name.

Step 3 - Declare 2 variable from the user (size, i) thereby followed by printf statement (used in size).

Step 4 - Use the for loop so that more than 1 records can be stored.

Step 5 - Now use printf function so that the user can give input for the initialized id, cpga & name.

Step 6 - Use 2 printf statement after for loop, one for the student's data & the other to represent data in a tabular form.

Step 7 - Again enter for loop statement so as to print more than 1 records. Now terminate the program.

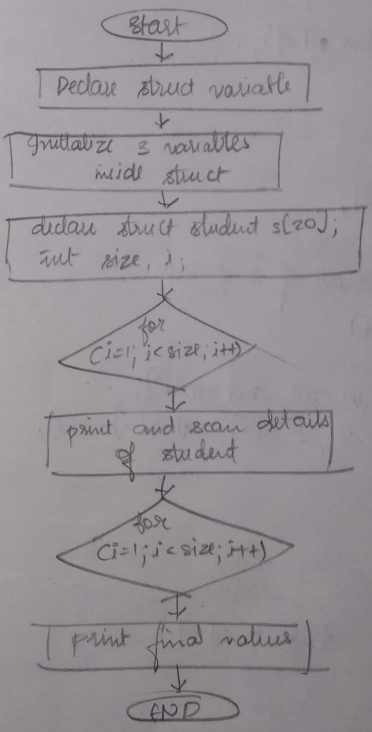
output -

How many records you want to insert : 2
0. Enter id : 1
0. Enter name : AK
0. Enter CGPA : 82.01%

1. Enter id : 2
1. Enter CGPA : 82.5%
1. Enter name : AL

```
{ printf ("\t%d \t%.f \t%.c")
}
return(1);
}
```

Flowchart -



Code -

```
#include <stdio.h>
#include <conio.h>
struct student
{
    int roll;
    char name[10];
};
void display (struct student s1[10]);
void main ()
{
  int i;
  struct student s[10];
  clrscr();
  printf("\n Enter details of 2 students:");
  for(i=0; i<2; i++)
  {
     printf("\n Enter roll and name");
     scanf("%d %s", &s[i].roll, s[i].name);
  }
 display(s);
 getch();

void display( struct student s1[10])
{
   int i;
   printf("\n ********** \n");
   for(i=0; i<2; i++)
       printf("\n Roll = %d \t Name = %s", s[i].roll, s[i].name);
}
```

B) Aim: WAP which will demonstrate use of structure & union.

Algorithm:

Step 1 - Start turbo C7 application.

Step 2 - Declare the structured variable as struct student

Step 3 - Initialize the struct student with two more variables (int roll and char name [10]).

Step 4 - Now inside void main display and void main declare int i, struct student s[10].

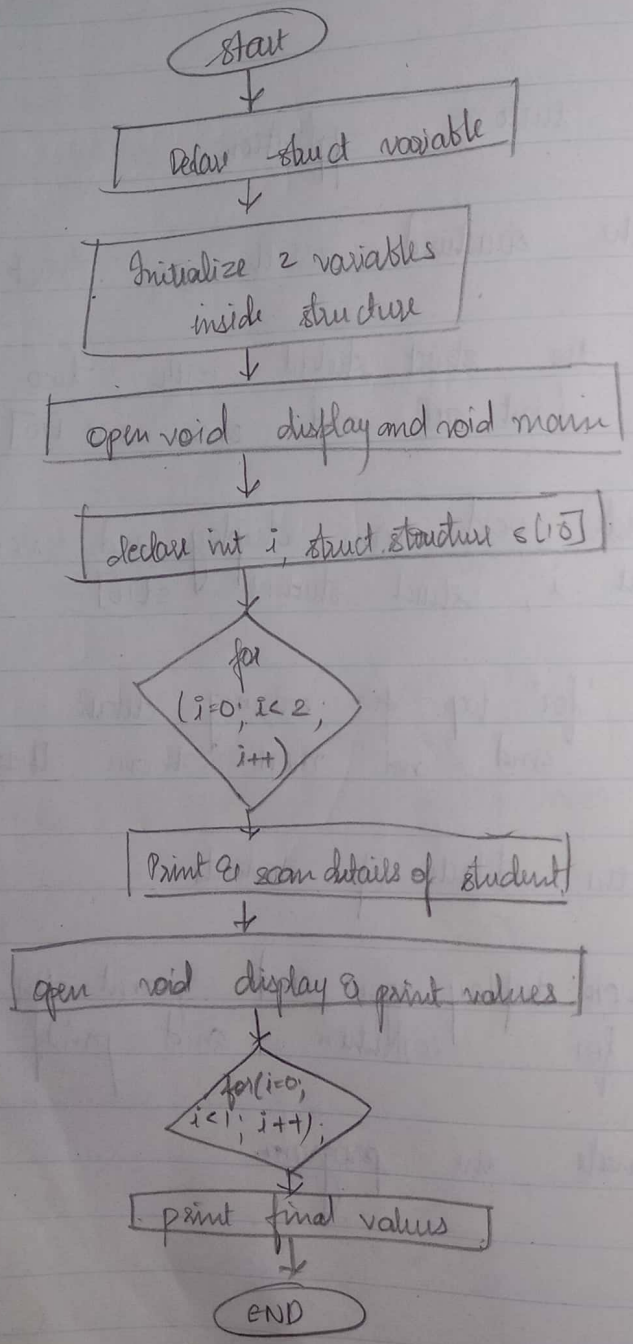
Step 5 - Use the 'for' loop for entering details of student upto 2 and not more than that.

Step 6 - Print the details of students.

Step 7 - Open void display again and print the values using for condition and printf.

Step 8 - Terminate the program.

Flowchart -



Output -

Enter details of 2 student
Enter roll and name 22 Ria
Enter roll and name 23 Trish
* * * * * * *
Roll = 22 Name = Ria
Roll = 23 Name = Trish

CODE -

```
#include <stdio.h>
#define MAX_SIZE 100
int main()
{
    char text1[MAX_SIZE];
    char text2[MAX_SIZE];
    int i;
    printf("Enter any string:");
    gets(text1);
    for(i=0; text1[i]!='\0'; i++)
    {
        text2[i] = text1[i];
    }
    text2[i]='\0';
    printf("First string = %s\n", text1);
    printf("First string copy = %s\n", text2);
    printf("Total characters copied = %d\n", i);
    return 0;
}
```


PRACTICAL - 9

Aim: WAP to copy one string into another string.

Algorithm:

Step 1- Input string from user and store it to some variable say text 1.

Step 2- Declare another variable to store copy of first string in text 2.

Step 3- Run a loop from 0 to end of string. The loop structure should be like.
for(i=0; text1[i] != '\0'; i++)

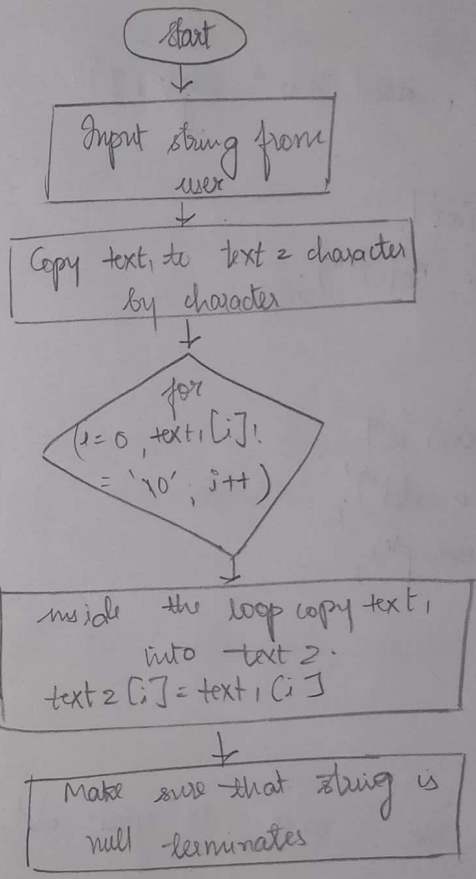
Step 4- Inside the loop for each character in text 1 copy to text 2. say text2[i] = text1[i]

Step 5- Finally after loop make sure the copied string ends with NULL character i.e text2[i]='\0';

Output -

Enter any string There are 7 days in a week
First string There are 7 days in a week.
First copy string There are 7 days in a week.

Total characters copied: 26

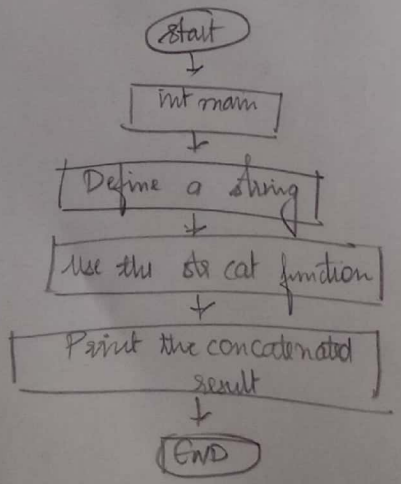


Code:

```
#include <stdio.h>
#include <string.h>
int main (int argc, const char * argv [])
{
    char example [100];
    strcpy (example, "Rahul");
};
strcat (example, " is over 18");
strcat (example, " years old");
printf ( "%s\n", example);
return 0;
}
```

Output : Rahul is over 18 years old.

Flowchart -



3. Aim : Write a program which will demonstrate the use of string library function.

strcat : The strcat() function will append a copy of the source string to the end of destination string. The strcat function takes 2 arguments: 1) dest
2) str

The strcat function returns a pointer (where the resulting concatenated string resides).

Conclusion - We have sucessfully executed the program.

C Aim: WAP which displays the length of a string without using string function.

Algorithm:

Step 1 - Take a string as input & store it in the array.

Step 2 - Using for loop count the number of characters in array & store the results in the variable.

Step 3 - Print the variables as output.

#CODE:

```
#include <stdio.h>
void main()
{
    char string[50];
    int i, length = 0;
    printf("Enter a string \n");
    gets(string);
    for(i=0; string[i] != '\0'; i++)
    {
        length ++;
    }
    printf("The len of str is the no. of characters in it");
    printf("so the len of : %s = %d \n", string, length);
}
```

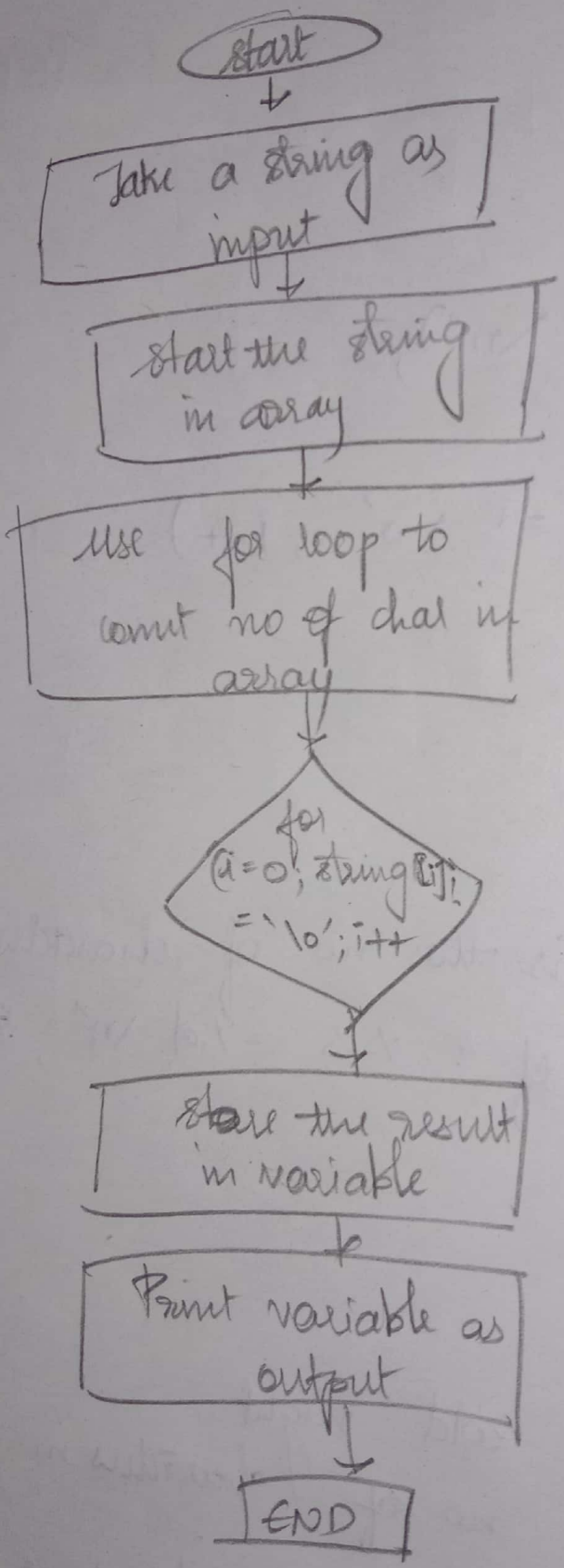
Output -

Enter a string It is a cold night

The length of str is the no of characters in it

so the length of : It is a cold night = 18.

820

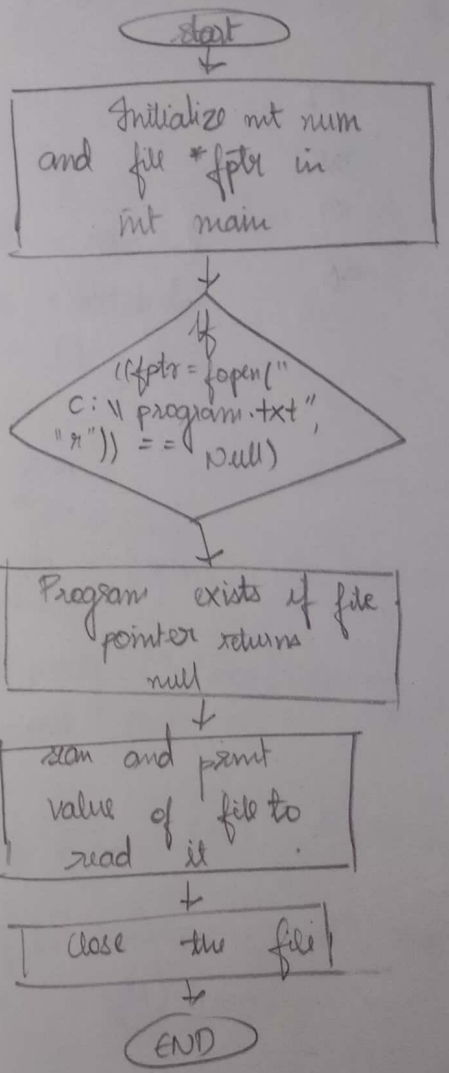


PRACTICAL : 10

Aim : Program for file open, file read and file close.

- fopen() → opens an existing file or create a new file for use
- fread() → Reads a record from a file.
- fclose() → closes a file.

Read from text file and close d



Output :

values are = 87
88
89
90

Code for reading from a text file opening / closing .txt file is and its contents are:

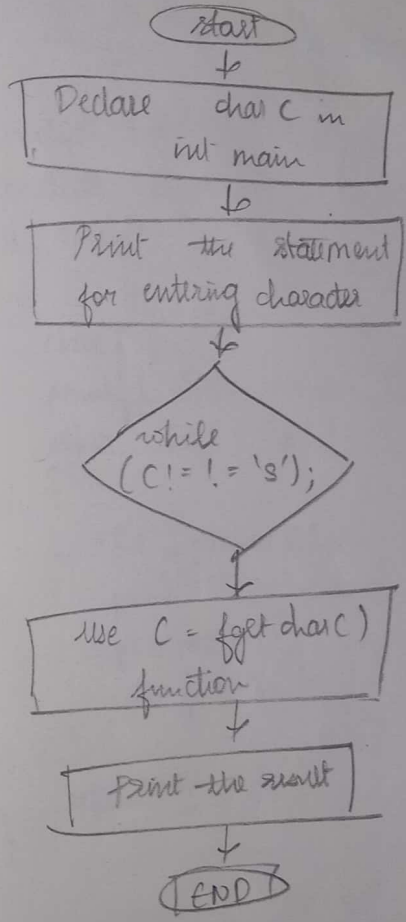
87
88
89
90

```
# include <stdio.h>
# include <stdlib.h>
int main ()
{
    int num;
    FILE *fptr;
    if ((fptr = fopen("C:\\program.txt","r")) = NULL)
    {
        printf ("Error! opening file");
        exit (1);
    }
    fscanf (fptr, "%d", &num);

    printf ("values are = %d", num);
    fclose (fptr);

    return 0;
}
```

Aim : WAP for fgetc(), fgetchar(), fputchar() functions.

Algorithm / Description

- fgetchar is a file handling function
- It is used to read a single character from keyboard input.



#Code:

```
#include <stdio.h>
#include <ctype.h>
int main()
{
    char c;
    printf("Enter some character. Enter to exit \n");
    while (c != '$');
    {
        c = fgetchar();
        printf("\n Entered character is : ");
        putchar(c);
        printf("\n");
    }
    return 0;
}
```

Output: Enter some character Enter $ to exit....

A

Entered character is: A

B

Entered character is : B

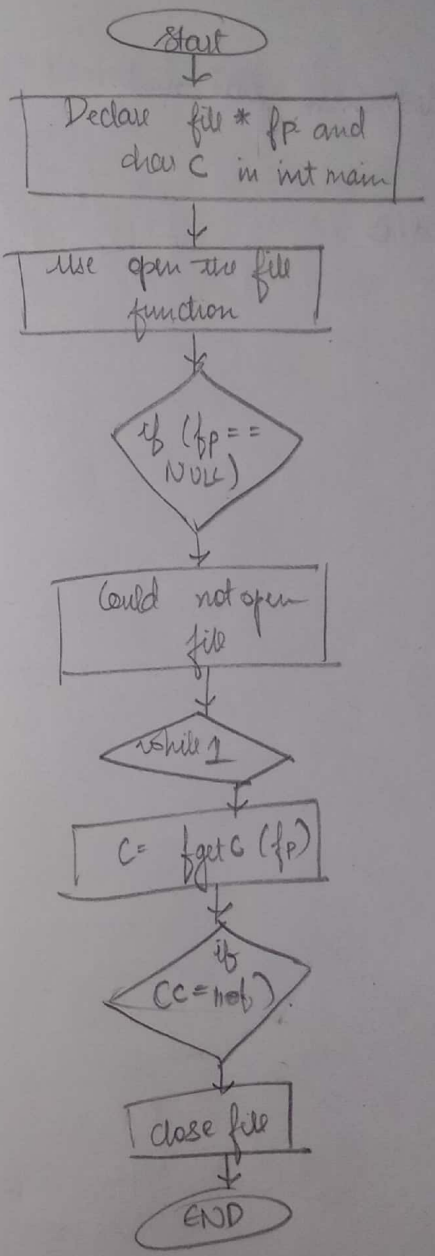
$

Entered character is: $

- fgetc() → Used to read a character from a file. Reads single character at a time.
in a program we use fgetc() function
fgetc(fp);
where
fp = file pointer

CODE:

```
#include <stdio.h>
int main()
{
 file *fp;
 char c;
printf("Opening file test.c in read mode");
fp = fopen("test.c", "r");
if (fp == NULL)
{
   printf("could not open file test:(");
   return 1;
}
printf("Reading the file test.c");
while (1)
{
   c = fgetc(fp);
   if (c == EOF)
   break;
   printf("%c", c);
}
```



Output :

```
Opening the file test.c in read mode
Hi, How are you?
Closing the file test.c
```

```
    printf ("Closing file test.c");
    fclose (fp);
    return 0;
}
```

fputchar() → · file handling function in C
· used to write a character on standard output / screen
· fputchar() function is equivalent to putchar() function where char is a character variable.

CODE :

```
#include <stdio.h>
int main()
{
    char ch = 'a';
    while (ch <= 'z')
    {
        fputchar(c);
        ch++;
    }
        return 0;
}
```

Output : abcdefghijklmnopqrstuvwxyz