

```
coding :-
1) Students details
# include <stdio.h>
# include <conio.h>
void main()
{
    int rollno;
    char name[20], mobileno[10];
    float percentage;
```

```
clrscr();
printf("Enter student's name: \n ");
scanf("%s", &name);
printf("Enter student's rollno: \n ");
scanf("%d", &rollno);
printf("Enter student's mobile-no: \n ");
scanf("%s", &mobileno);
scanf("%f", &percentage);
```

```
printf("Enter student's percentage: \n ");
scanf("%f", &percentage);
printf("Student's name: %s\n", name);
printf("Student's rollno: %d\n", rollno);
printf("Student's mobile-no: %s\n", mobileno);
printf("Student's percentage: %.2f\n", percentage);
getch();
```

Aim:- Write a C program to understand the basic datatype and input output.

Requirement :- Turbo C++, Notepad

Algorithm :-

1. Declare a variable integer to store roll number a float variable to store percentage and two string array for storing name and variable were numbers.
2. Print using the printf() method to get for values and we scanf() to get the required values

3. After running the values, print them out by one using the printf() statement.

Output:-

Enter your name : Rakesh

Enter your rollno : 1868

Enter your mobileno : 80997882456

Enter your percentage : 60%.

* Conclusion:-

The integer, character and float datatype has been studied.

Name

Practical-2

30

```

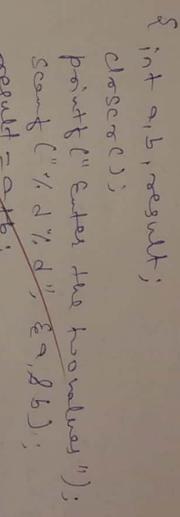
coding :-  

# include <stdio.h>  

# include <conio.h>  

void main()
{
    int a, b, result;
    clrscr();
    printf("Enter the two values:");
    scanf("%d %d", &a, &b);
    result = a+b;
    printf("a+b=%d", result);
    result = a-b;
    printf("a-b=%d", result);
    result = a*b;
    printf("a*b=%d", result);
    result = a/b;
    printf("a/b=%d", result);
}

```



- Q) Expressions (Dynamic):-
- Algorithm:-
1. Declase three variable . Two to accept two values and one to store result .
 2. use the scanf() function to accept two values from the user.
 3. use the various operators such as ('+', '-', *) etc. to perform arithmetic operation on the variables and store the value in the third variable and print it .

Conclusion:-

The various expression and binary operations have been studied.

Output:-
Enter the two values = 15, 10

$a+b = 25$
 $a-b = 5$
 $a*b = 150$
 $a/b = 5$

Coding :-
~~#include <conio.h>
include <stdio.h>~~
void main()

```
{ int a,b,c;
printf ("Enter any two numbers");
scanf ("%d,%d", &a,&b);
c=a>b ? a : b;
printf ("The greater number is %d", c);
getch();}
```

Output:-

Enter any two numbers : 15,20

The greater number is 20.

④ Ternary

Algorithm :-

1. Declare 3 integer variables
2. Use the scanf() method to accept the values.
3. Use the ternary operator to store the value of the greater number in the third variable.
4. Print the greater number.

Conclusion:-

The ~~ternary~~ operator works like a conditional statement if-else and stores the resultant value in a desired variable.

~~Notes~~

18 Practical 3

32

Coding :-
#include <iostream.h>
#include <conio.h>
void main()

{
int a;
clrscr();

cout<"Enter an integer";
scanf("%d", &a);

if (a > 15)

 cout<"The entered number is greater than 15";

 else
 cout<"This block is out in the if section";

getch();

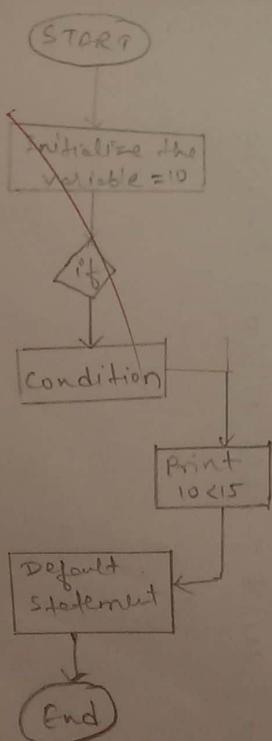
}

- Q 1. If statement :-
- a) Algorithm :-
Take a integer input after declaring an integer variable using the scanf () method.
 - b) If the given integer is greater than 15 print that it is greater than 15 using the if condition statement.

- Q 2. Also outside of the if block print a statement indicating that it is outside the scope using the if statement.

Conclusion:-
The if condition statement executes a block of code only if a certain condition is true.

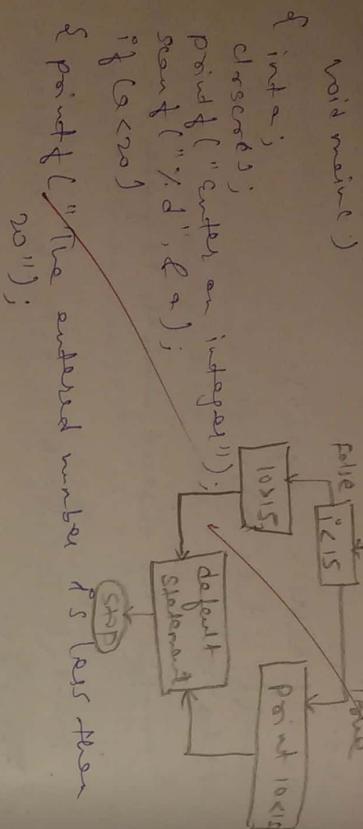
Output :-
Outer one integer: 18
The entered number is greater than 15
This block is out of the if statement.



38

Initialize the
variable $x=20$

```
Coding:-  
# include <stdio.h>  
# include <conio.h>  
void main()
```



③ If - else Statement :-

Algorithm :-

- 1. Initialize an integer variable . use the scanner method to take the input from the user.
- 2. we use if statement to check whether number is less than 20 . if so, print it is less than 20.

2.

- 3. we use also statement to check whether the number is greater than 20 and print a statement accordingly .

4. terminate the program .

Conclusion :-

The 'else' part is executed , the condition of the if statement is not met .

Output:-
Enter an integer: 18
The entered number is less than 20 .

39

(iii) Nested if

Algorithm:-

1. Take an integer, then we use class `Scanner` method to clear the console.
2. we use `if` conditional statement to check whether the given number is greater than or equal to 50. we another `if` statement inside the block to check whether the input is equal to 50. we use `else` statement to point the response that the number is greater than 50.
3. we use `else` statement to point that the number is smaller than 50.

Conclusion:-

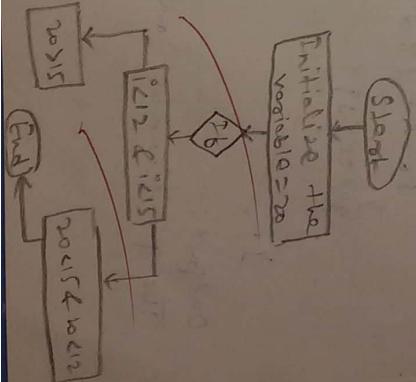
Here, multiple `if` conditional statement are needed together.

Coding:-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a;
    clrscr();
    printf("Enter an integer : ");
    scanf("%d", &a);
    if (a == 50)
        {
            printf("Number is 50");
        }
    else
        {
            printf("Number is less than 50");
        }
}
```

Output:-

Enter an integer: 50
 Number is 50.
 Enter an integer: 60
 Number is greater than 50.



```
#include <stdio.h>
#include <conio.h>
```

```
void main()
```

```
{ int n, i, j;
```

```
char str[5];
```

```
point("The prime number are: ");
```

```
for (i = 2; i <= 20; i++)
```

```
{ do
```

```
for (j = i + 1; j <= n; j++)
```

```
{ if (i * j == 0)
```

```
{ att;
```

```
str[i] = '0';
```

```
att;
```

```
y = 1;
```

Output:

The prime numbers are: 2, 3, 5, 7, 11, 13 ...

a) Aim:- To display the prime number using for loop.

Algorithm:

Step I :- Initialise the variable out of which two are loop variable and one is count variable.

Step II :- Initialise a for loop from 1 to 50 let the count variable be zero.

Step III :- Nest another loop within the loop in Step 2 that goes to 2 to the first

variable var

Step IV :- we use the if condition statement to check

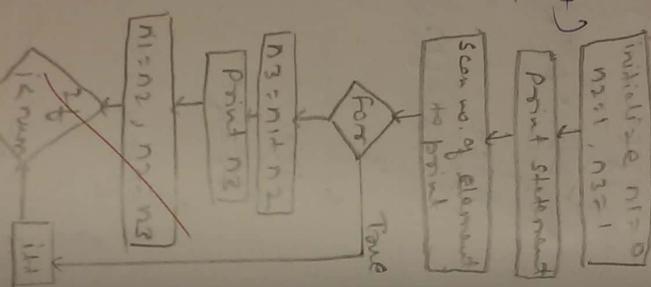
whether first loop variable 1st 2nd loop variable - 10, it has increment count variable by 1.

Step V :- come out of the second loop & check whether the count variable is 0 if true

print the number.

Step VI :- terminate the program.

Conclusion:- Thus, we have successfully execute prime number on turbo C.



b) Write a program of fibonacci series.

program:-

step1: Start the Turbo C

Step2: Declare the variable n1, n2, n3, i, number

steps:- Initialize the variable n1=0, n2=1 and number=0

Step3:- Enter the number of term of fibonacci series to be printed.

Step4:- Point first two term of series as n1=0, n2=1

Step5:- we a for loop as per following step.

n1=n1+n2

n2=n2

n2=n3

Successive value of 1 element each time by 1

Step6:- print the values of number.

Step7:- End

Conclusion:- Thus, we were successful execute fibonaci series on Turbo C

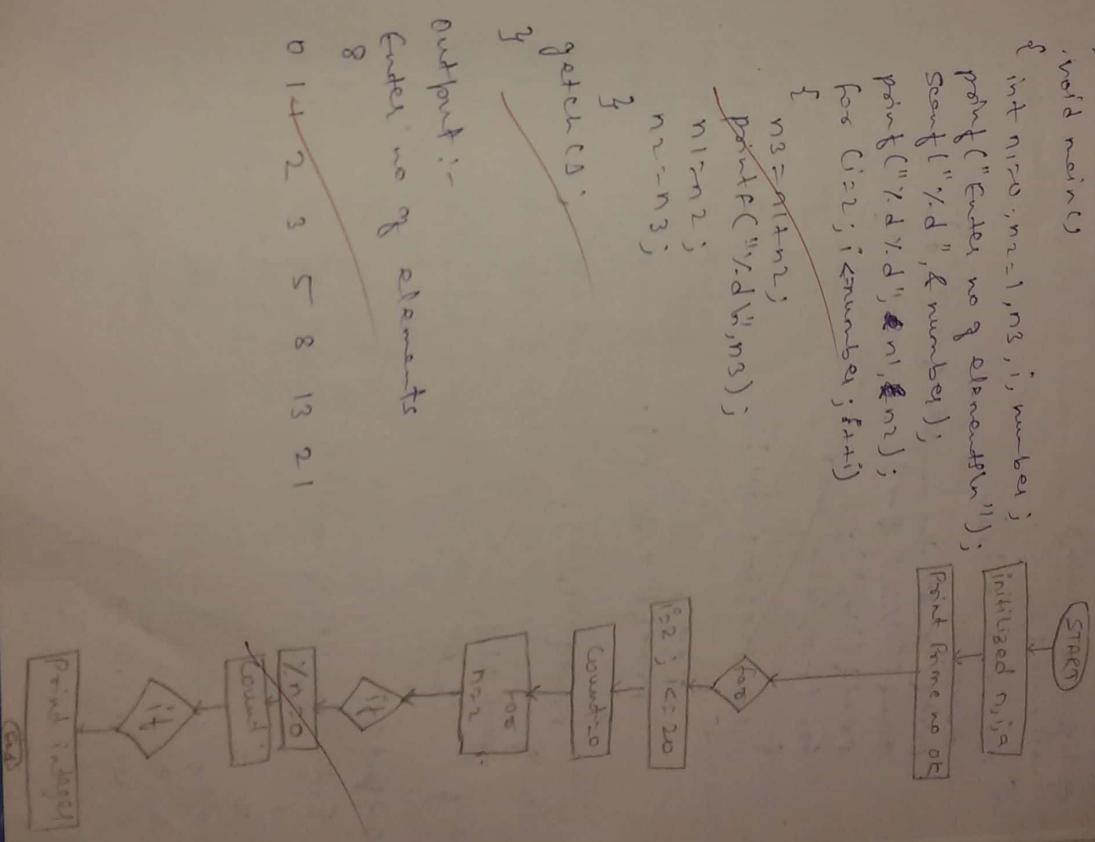
contd:-

#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("1 & 2\n");
    for (i=2;i<number;i++)
    {
        n3=n1+n2;
        printf("1 & %d\n",n3);
        n1=n2;
        n2=n3;
    }
}
```

36



include <stdio.h>

include <conio.h>

void main()

```
{ int n=0, i, j;
```

```
class();
}
```

```
point("Enter the number of rows: ");
```

```
scanf("%d", &n);
```

```
point("In ");
```

```
for (i=0, j=n-1;
```

```
for (j=0; j<i; j++)
    for (i=0; i<n; i++)
        for (j=0; j<i; j++)
            point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

```
for (i=0; i<n; i++)
    for (j=0; j<i; j++)
        point("In ");
```

Start

Initialize the n=0

Print no of rows

Scan the element

for

for

for

for

for

for

for

for

for

End

- c) Write a program on following expression.

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

Algorithm:

Step 1: Start the Turbo C program.

Step 2: Declares the variable rows, i, j, number=1

Step 3: Display the number of rows

Step 4: Enter the for loop at i=1; i<rows; i++
Step 5: Now create nested for loop at j=1; j<i; j++
Step 6: Display the number as per user enter the sequence from i-1.

Output:
Enter number of rows: 4
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

Step 1: Increment number from 1.
Step 2: Display the space
Step 3: End.

Conclusion: - Thus we successfully execute given expression in Turbo C using nested for loop.

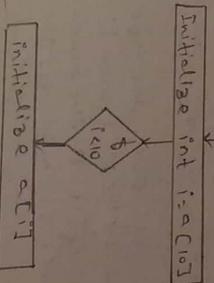
Wrote

Practical 5

Code :- A

38

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[10], i;
    clrscr();
    printf("Enter the elements in list");
    for (i=0; i<10; i++)
        {
            scanf("%d", &a[i]);
            if (a[i] < a[0])
                {
                    a[0] = a[i];
                }
        }
    printf("The largest no is %d", a[0]);
}
```



A
Aim:- A program to find longest array number using array.

* Algorithm:-

Step 1: Start Turbo C application

Step 2: Define the variable i and integer array a[10]

Step 3:- Enter the for loop at i=0, i<10 and we find value of a[0] till i<10 .
exit for loop.

Step 4:- Enter the for loop at i=0, i<10. we if condition statement to check if a[0] less than print put a[0]=a[i]

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

Step 6:- terminate the program .

Conclusion:- Thus we executed successfully
executed longest number using
array in Turbo C

The longest no. is 10.

```

code - B
#include <stdio.h>
# include <conio.h>

void main()
{
    int array[100], i, num;
    clrscr();
    cout << "Enter the size of array ";
    cin >> n;
    cout << "Enter the elements of array ";
    for (i = 0; i < n; i++)
        cin >> array[i];
    cout << "Enter the element to be searched ";
    cin >> num;
    for (i = 0; i < n; i++)
        if (array[i] == num)
            cout << "Element found at index " << i;
    getch();
}

```

Aim: write a program to print the number of odd & even no. in the array.

* Algorithm:

Step 1: Create an array take its size from user and define its elements using loop.

Step 2: Display the size of array.

Step 3: Display the elements of array entered by the user.

Step 4: Take the initiation in for loop using which all the elements of array exist.

Step 5: Display even no from the array for for loop.

Step 6: - If $\text{array}[i] \neq 0$
display the even no. from given array.

Step 7: Display the odd no. from array
if $\text{array}[i] \neq 0$

Step 8: Close the for loop c.

Conclusion: Thus, we successfully execute odd & even no. in turbo C.

Output:-

40

Enter the size of array

5

Enter the element of array

39

5

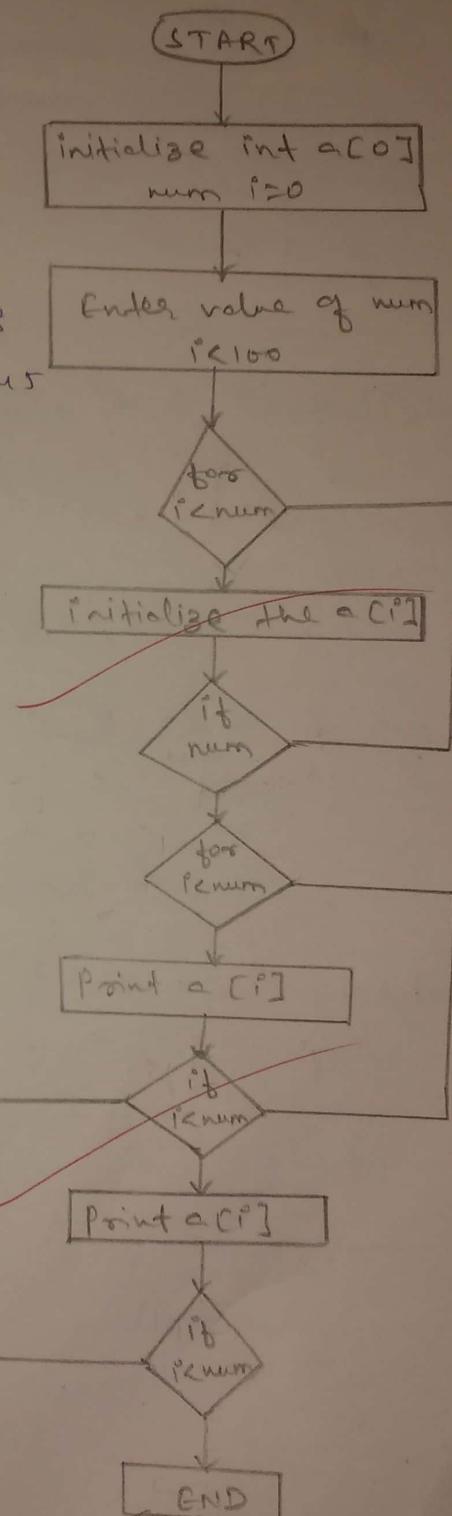
2678

123

45

Even no. in the array : 342678

odd no. in the array : 5 123 45



Aim:- C program to find ~~avg~~ and sum using arrays.

* Algorithm:-

Step 1:- Start Turbo C.

Step 2:- Declare the int variable n, i, initializing num[100], sum=0.0, avg.

Step 3:- Using for loop at i=0; i<n; i++. give input user age and increment i by 1.

Step 4:- Declase sum variable and store it by adding num[i]

Step 5:- Average is sum divided by n.

Step 6:- Given point statement for average and sum.

Step 7:- Terminate the program.

* conclusion:-

Thus, we have executed the program successfully.

Yours,

Code :-

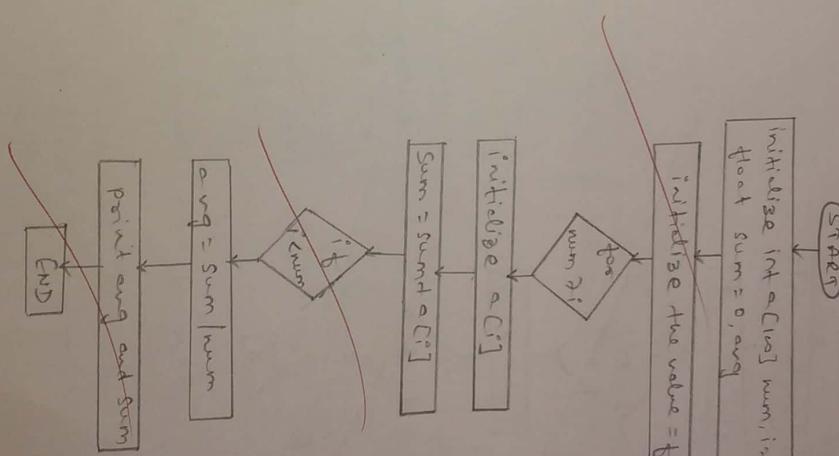
```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n,i;
    float num[10], sum=0.0, avg;
    clrscr();
    printf("Enter the no. of element ");
    Scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        printf("Enter no %d ", i+1);
        scanf("%f", &num[i]);
        sum = sum + num[i];
    }
    avg = sum/n;
    printf("Sum = %f", sum);
    getch();
}
```

Output :-

Output after the no. 9 element.

42

0 1 2 3 4 5 6 7 8 9 10



A. Aim is factorial of a number using recursion.

Algorithm:-

Step 1:- Start the Turbo C application.

Step 2:- Declare a function prototype returning the main() function.

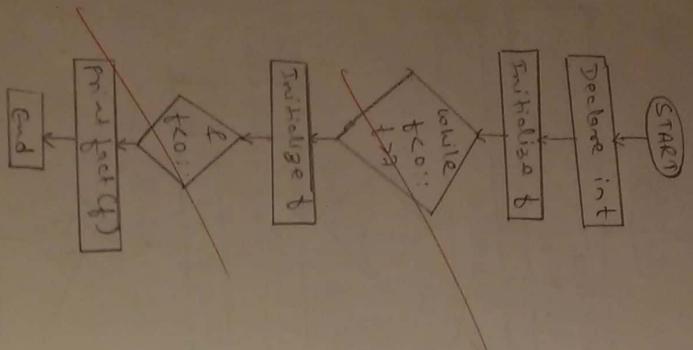
Step 3:- Declare a variable 'f' in the main() function.

Step 4:- Use the while function entered by the user to execute that entered value is in range 0 to 17.

Step 5:- Print the value entered by the function in step-2.

Step 6:- Terminate the program.

Conclusion:- Thus, we successfully execute calculating a number using recursion in Turbo C.



Code-A

```
#include <stdio.h>
#include <conio.h>
void main()
```

```
{
```

```
int f;
```

```
clrscr();
```

```
pointf("Enter the no. to find the factorial  
of (0 to 12)");
```

```
scanf("%d", &f);
while (f < 0 || f > 12)
```

```
{
```

```
pointf("Y/D", fact(f));
scanf("%c", &f);
```

```
if (f == 'Y' || f == 'y')
```

```
pointf("Y/D", fact(f));
getch();
```

```
{
```

```
int fact(int n)
```

```
{
```

```
if (n >= 0)
```

```
{
```

```
return n * fact(n - 1);
```

```
else
```

```
{
```

```
y
return 1;
```

Output:-

Enter the no. to find factorial of (0 to 12) = 6
4! 720

4!

B Aim:- Write a program which shows the use of getch() function.

Algorithm:-

Step 1:- Initialize the Turbo C application.

Step 2:- Initialize the character variable 'ch'.
Step 3:- Use the getch() method to accept the character.

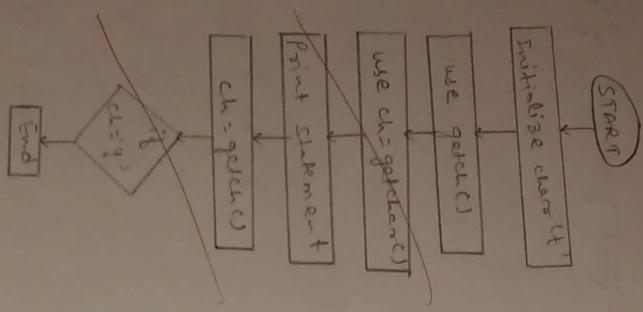
Step 4:- Use the getch() method to show an option 'y/n' in 'ch'.

Step 5:- Which 'ch' keep accepting value by 'ch'.

Step 6:- Use the puts() method to show 'y', 'n' in 'ch'.

Step 7:- Terminate the program.

Conclusion:- Thus, we successfully execute the use of getch() function in Turbo C



47

Code - 8

```
#include <stdio.h>
#include <conio.h>
void main()
```

```
{ char ch;
```

```
clrscr();
```

```
putch("Press any key to continue");
```

```
getch();
```

```
clrscr();
```

```
printf("Would you like to continue Y/n");
```

```
ch = getch();
```

```
while(ch == 'y')
```

```
{ printf("would you like to continue") ;
```

```
ch = getch();
```

y.

Output:-

Press any key to continue
 Enter any character : A
 would you like to continue ? - y
 would you like to continue ? - n

48

Point w.r.t to show the use of put() function.

Algorithm:

Step 1:- Start the Turbo C application.

Step 2:- Initialize a character 'c' to 'b'.

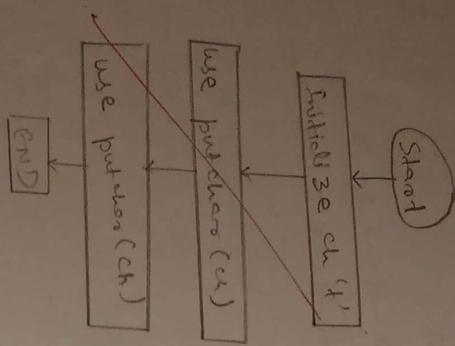
Step 3:- use the puts() and putchar() function with 'c' as the argument.

Step 4:- Terminate the program.

~~Conclusion~~

The factorial uses of getch and puts() function is above successfully.

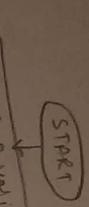
~~Chmod~~



Code - C

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char ch = 'A';
    clrscr();
    putch(ch);
    printf("\n");
    putch(ch);
    getch();
}
```

Output:-
A
A



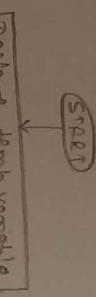
Aim :- Swapping of array using pointers.

Algorithm:-

Step 1 :- Start the Turbo C application.

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

Swap C program



Step 3 :- Declare 2 variable and accept their value from the user, point the &pushing value using printf().

Step 4 :- Take the address of the variable as argument for the function.

Step 5 :- Point the respective value of the variable.

Step 6 :- we use the basic swapping algorithm in the function definition but instead of normal variable use,

Conclusion :- Thus we have executed program successfully.

Code A

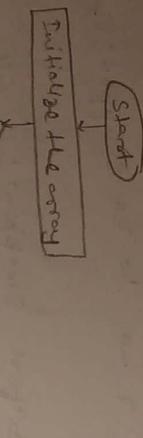
```
#include <stdio.h>
#include <conio.h>
void main()
{
    int x,y;
    clrscr();
    cout("Enter the two number to be swapped:");
    swap(x,y);
    cout("x = " + x + " and y = " + y);
    cout("The value before swapping: " + x + " " + y);
    cout("The value after swapping: " + y + " " + x);
}
```

Output:-

Enter the 2 no. to be swapped: 12 24

The no. before swapping: 12 24
The no. after swapping: 24 12

50



B
Bubble sorting of array using pointer.

Algorithm:-

Step 1:- Initialize an integer array and temp variable.

Step 2:- Run a nested loop of i=0 (arr[a]) and
of j=0 to len(a)-1.

Step 3:- If arr[j] > arr[j+1] swap the two values using
basic swapping logic

Step 4:- print the swapped array.

Step 5:- Terminate the program.

Conclusion:- Thus we have executed program
successfully.

Code - B

```

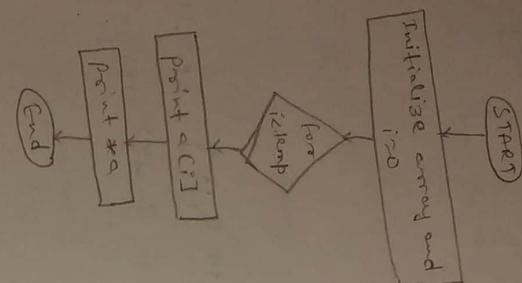
void sort (int arr[], int n)
{
    #include < stdio.h>
    #include < conio.h>
    void main()
    {
        int a[10], i, j, temp;
        clrscr();
        for (i = 0; j < n; i++)
        {
            for (j = 0; j < i; j++)
            {
                if (a[j] > a[i])
                {
       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

```

Output is

Insert elements into the array

1 2 3 4 5 6 7 8 9 10 11 12
 { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 } is the
 Sorted array



Aim:- One dimensional array representation using pointers.

Program:-

Step 1:- Start the Turbo C algorithm.

Step 2:- Initialise an integer array and a variable.

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

Step 4:- Point the data of the array and then use pointer to print array location.

Step 5:- Terminate the program.

Conclusion:- Thus, we have executed program successfully.

coding:-

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int a[5] = {7, 9, 4, 8, 13}
    clrscr();
    int *ptr;
    int i;
    ptr = &a[0];
    while (*ptr != '\0')
    {
        printf("The address of a[%d]\n"
               "= %u", i, ptr);
        printf("The value of a[%d]\n"
               "= %u", i, *ptr);
        ptr++;
    }
}
```

Output:-

The address of a[0] = 65516
 The value of a[0] = 7

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

Aim:- Various operation on struct.

1) Structure Student.

Algorithm:-

Step 1:- open the Turbo C application.

Step 2:- Define a Struct 'Student' under id, class and name variable.

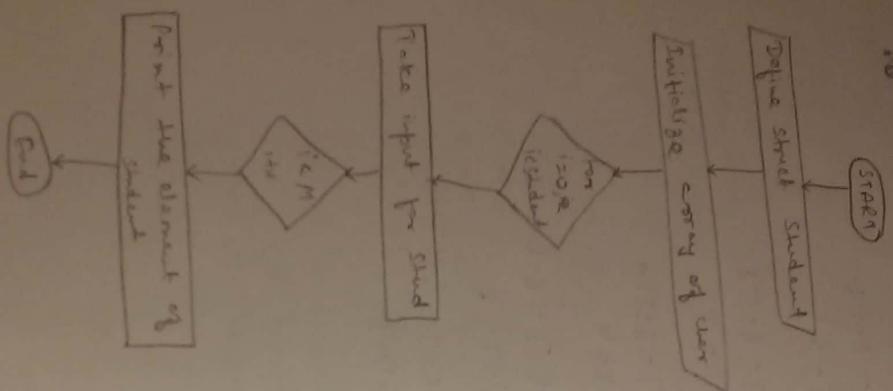
Step 3:- In the main function, declare an array of Student.

Step 4:- we use the scanf() function to scan the various element of the student element.

Step 5:- Print all the elements of the student structure using printf for loop

Step 6:- Terminate the program

Conclusion:- we have executed program successfully.



```

Coding :-  

# include <stdio.h>  

# include <conio.h>  

struct student  

{  

    int id, class;  

    char name[20];  

};  

void main()  

{  

    struct student s[2];  

    int i;  

    clrscr();  

    for(i=0; i<2; i++)  

{  

    printf("Enter details of student %d ", i+1);  

    scanf("%d", &s[i].id);  

    scanf("%c", &s[i].class);  

    scanf("%s", &s[i].name);  

}  

    for (i=0; i<2; i++)  

{  

    printf("The details of student %d ", i+1);  

    printf("ID : %d Name : %s", s[i].id, s[i].name);  

    printf("Class : %c", s[i].class);  

    printf("\n");
}
}

```

Enter the detail of Student 1

1868

12

Rahul

Enter the detail of Student 2

1839

12

Tushar

Detail of Student 1

ID : 1868

Class: 12

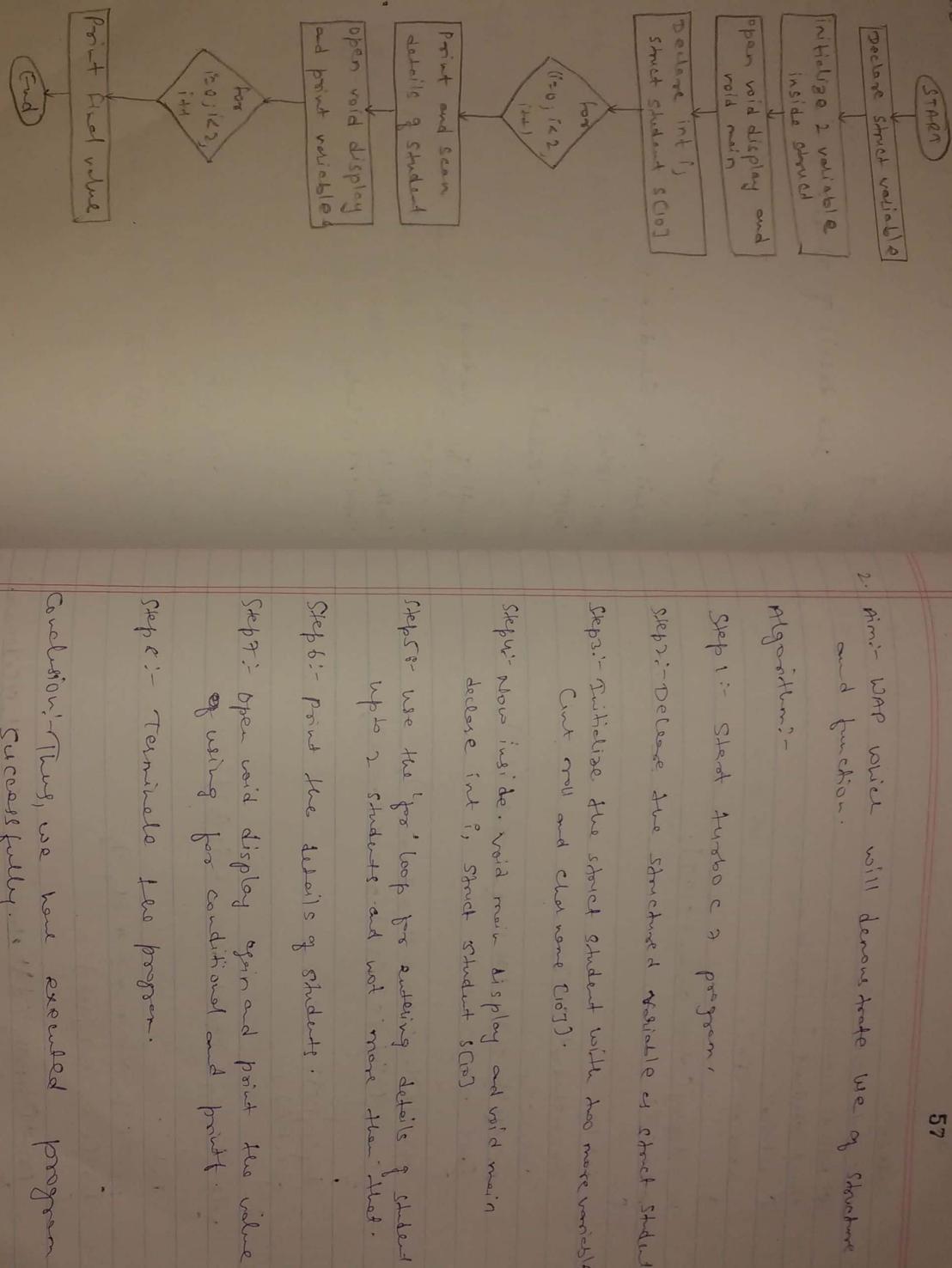
Name: Rahul

Detail of Student 2

ID : 1839

Class: 12

Name: Tushar



2. Aim:- WAP which will demonstrate use of Structure and function.

Algorithm:-

Step 1:- Start Turbo C program.

Step 2:- Declare the structure variable & struct student.

Step 3:- Initialize the struct student with two more variable Cint roll and char name (roll).

Step 4:- Now inside void main display and void main declare int i, Struct student s[0].

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

Step 6:- Print the details of students .

Step 7:- Open void display again and print the value by using for conditional and print .

Step 8:- Terminate the program.

Conclusion:- Thus, we have executed program successfully.

Output:

Enter details of student.

Enter roll and name 22 Rohit

Enter roll and name 24 Twinkle

* * * * *

Roll = 22 Name = Rohit

Roll = 24 Name = Twinkle

Coding:-

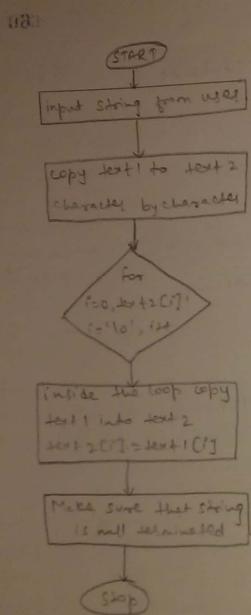
```

#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 s1[10];
    clrscr();
    printf("Enter details of 2 students ");
    for(i=0; i<2; i++)
    {
        printf("Enter roll and name ");
        scanf("%d %s", &s1[i].roll, s1[i].name);
        display(s1);
        getch();
    }
}

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

```



Aim:- WAP to copy one string into another string

Algorithm:-

Step1:- Input String from user and store it to some variable say text1.

Step2:- Declare another variable to store copy of first string in text2.

Step3:- Run a loop from 0 to end of string. The loop structure should be like
for(i=0; text1[i] != '\0'; i++)

Step4:- Inside the loop for each character in text1 copy it to text2. say text2[i] = text1[i].

Step5:- Finally after loop make sure the copied string ends with NULL character i.e.
text2[i] = '\0'.

Conclusion:- Thus, we have executed program successfully.

Coding:-

```
#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 character copied = %d\n", i);

    return 0;
}
```

Output :-

Enter any string: There are 7 days in a week

First string: There are 7 days in a week

First string copy: There are 7 days in a week

Total characters copied: 26.