

Practical-01

28

Ques: Write a C program to understand the basic data types.

Algorithm:

Step I: Declare a variable ~~integer~~ to store roll number.

A float variable to store Percentage and two

things array for storing name and mobile number

Step II: Print using "print f" method to ask value and use "scanf" to store the value.

Step III: After receiving the values print them by using the "printf()" statement.

→ Conclusion:

Thus the integer, character and float data type
been studied.

Output:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int roll;
    float perc;
    char name[40], mob[12];
    clrscr();
    printf("Enter your name:");
    scanf("%s", &name);
    printf("Name: %s", name);
    printf("Enter your roll no:");
    scanf("%d", &roll);
    printf("Roll No: %d", roll);
    printf("Enter your Percentage:");
    scanf("%f", &perc);
    printf("Percentage : %f", perc);
    printf("Enter your mob no.:");
    scanf("%s", &mob);
    printf("Mobile No: %s", mob);
    getch();
}

→ Enter your name : Smit
Name: Smit
→ Enter your roll number: 1862
Roll No: 1862
→ Enter your Percentage : 88.77
Percentage : 88.77
→ Enter your mob no : 1234567890
Mobile no.: 1234567890
```

Practical - Q2

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

```
getch();
```

Ques:- Enter the two values :- 15

10

a+b=25

a/b=1

a*b=150

a%b=5

Start

Enter 1st number

Enter 2nd number

Print the result
in the third variable
with all arithmetic
operations(+,-,*,
/,%).

END

Algorithms:

- (1) Declare three variables, two to accept value and one to store result.

- (2) Use the "scanf()" function to accept two values from the user

- (3) Use the various operator and '+', '-', '*', '/' to perform arithmetic operation on the variables and store the result in the variable and print it.

⇒ Conclusion:-

The various expression and binary operators have been studied.

iii) Ternary Operator:

Algorithm:

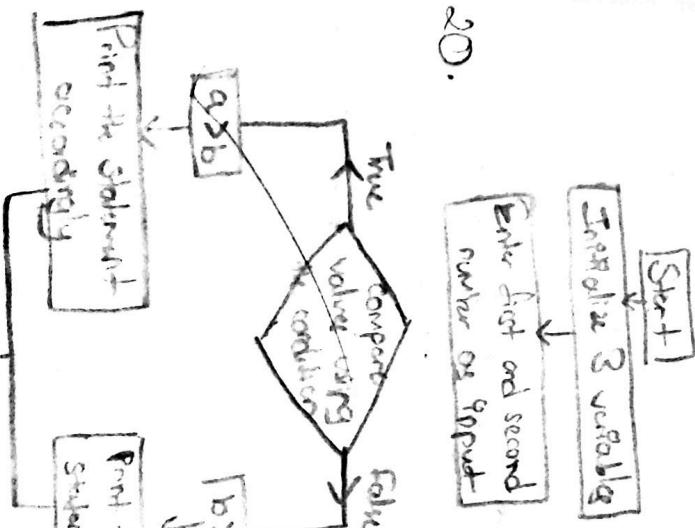
2. Step I: Declare 3 integer variable
3. Step II: Use the `scanf()` method to accept the value
4. Step III: Use the binary operator to store the value of the greater number in the third variable.
5. Step IV: Print the greater number

Conclusion:

The ternary operator work like an conditional if else and store the result in the respective variable.

Output:
Enter an integer: 15
20

The greater number is 20.



```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, x;
    clrscr();
    printf(" Enter an Integer: ");
    scanf("%d,%d", &a, &b);
    if (a > b ? a : b);
    printf("The greater number is ", a);
    getch();
}
```



```

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

```

```
void main()
```

```
{
```

```
int a;
```

```
clrscr();
```

```
printf("Enter an integer");
```

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

```
if (a > 15)
```

```
printf("Entered number is greater than 15");
```

```
else
```

```
printf("This block is outside the statement");
```

```
 getch();
```

```
}
```

Output: Enter an integer: 17

The entered number is greater than 15

This block is outside the statement

Practical 03:-

Conclusion:-

If the given integer is greater than 15, print that output is greater than 15 using the if conditional statement. Also outside of the 'if' block print a statement to indicate that output is outside the scope of statement.

Conclusion:-

Then if statement execution a block of code, only if a certain condition is satisfied.

else:-

Step 5: Initialize an integer variable with the `scanf()` method, and the input from the user.

In Step II: Use the if statement to check whether the number is less than 20 and then print a statement.

Step III: Examine the program.

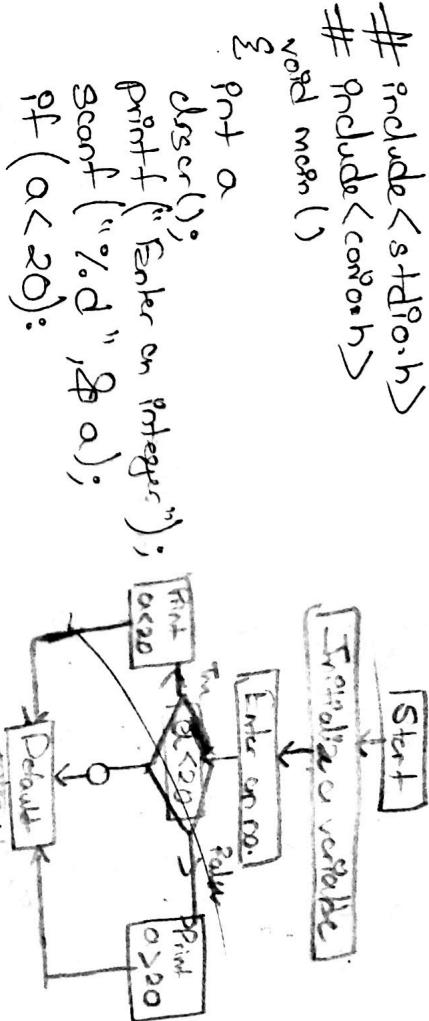
Conclusion -

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

```
        }  
    }  
    else  
    {  
        printf("The no is greater than 20");  
    }  
    getch();
```

~~Output:~~ Enter an integer : 17
The entered no. is 17

~~The entered no. is less than 20.~~



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

```
void main()
```

```
{
```

```
int a;
```

```
clrscr();
```

```
printf("Enter a number");
```

```
scanf("%d",&a)
```

```
if(a==80)
```

```
{
```

```
printf(a==80)
```

```
else if(a>80)
```

```
{
```

```
printf("a>80")
```

```
else
```

```
{
```

```
printf("The no. is less than 80");
```

```
else
```

```
{
```

```
printf("The no. is greater than 80");
```

```
}
```

```
getch();
```

```
}
```

```
Output - Enter an integer: 80
```

```
No. is 80
```

39

```

graph TD
    A[Enter a number] --> B{a>80}
    B --> C[Print a>80]
    C --> B
    B --> D{a=80}
    D --> E[Print a=80]
    E --> D
    D --> F{a<80}
    F --> G[Print a<80]
    G --> F
    F --> H([END])
  
```

Nested if:-

1. Declare an integer then we use the `clrscr()` function to clear the previous output.

2. Use the `if` conditional statement to check whether the given number is greater than or equal to 80. Again use the `if` statement inside the block to check whether the integer is equal to 80, then use the `else` statement to print the response that the number is less than 80.

3. Use the `else` statement to print that the num is less than 80.

Conclusion:-

These multiple `if` conditional statement are nested together.

Nested

Practical 04

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

Aim: To display prime numbers using for loop

Algorithm:

Step I: Initialize three variable out of which two are loop variables and one is count variable.

Step II: Initialize a for loop from 2 to 20 (value for the loop variable).

Set the count variable to zero.

Step III: Next another loop within the loop in Step II that goes from 2 to the first loop variable i.e. 2.

Step IV: Use the if-conditional statement to check whether (loop variable == 2) loop variable == 0) If true increment count variable by 1.

Step V: Come out of the second loop and check whether the count variable is 0. If true print the number (loop variable).

Step VI: Terminate the program

→ Conclusion:-

Prime Numbers were displayed using for-loop.

Output: The prime numbers are:

2

3

5

7

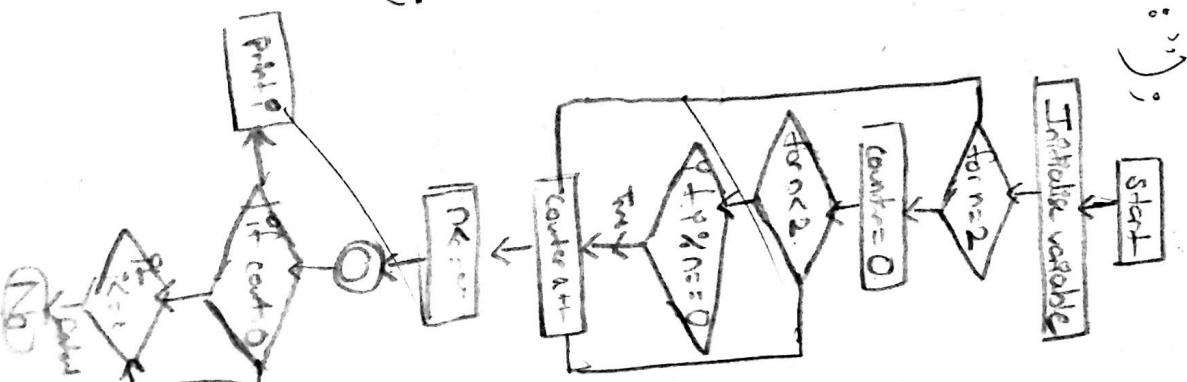
11

13

17

19

```
int n, a;
clrscr();
printf("The prime numbers are : ");
for (n=2, a=2; n<=20, a++)
{
    if (n%a==0)
        a++;
}
printf ("%d\n", n);
```



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

Input $n_1 = 0, n_2 = 1, p$, number, n_3 ,

Scan no. of elem to print

Algorithm

b)

b) Ans: Write a C program on Fibonacci series:

```
discr();
printf("Enter no. of elements : ");
scanf("%d", &number);
printf("%d %d", &n1, &n2);
```

```
for (p=2; p<=number; p++)
    for (i=2; i<=p; i++)
        n3 = n1 + n2;
```

```
    printf("%d", n3);
    n1 = n2;
    n2 = n3;
```

- I. Start a Turbo C application
- II. Declare the variables $n_1, n_2, n_3, i, number$
- III. Initialize the variable $n_1 = 0, n_2 = 1, number = 0$
- IV. Enter the number of term of Fibonacci series to be printed

V. Print first two numbers

VI. Print the first two numbers of series $n_1 = 0, n_2 = 1$

VII. Use the for loop as per following steps:

$$\begin{aligned}n_3 &= n_1 + n_2 \\n_2 &= n_1 \\n_1 &= n_2\end{aligned}$$

VIII. Then increase value of ' i ' element each time by 1

IX. Print the value of number

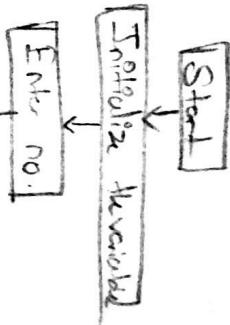
X. Terminate the program

Conclusion: Thus we have successfully executed Fibonacci Series on Turbo C7.

Q) Write C program on following expression

Algorithm Steps: Start Turbo C7 application

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int r, j, i, num;
    clrscr();
    printf("Enter the number of rows:");
    scanf("%d", &r);
    printf("\n");
    for (i=1, j=1; i<=r, j<=i; j++)
    {
        for (j=1, i<=r, j++)
        {
            printf("%d", j);
            if (j==i)
                break;
        }
        printf("\n");
    }
}
```



Step I: Decler variable r, j, i and use clrscr();

Step II: Display the number of rows

Step III: Enter the for loop ($i=1, j=1, j++$)
Create a nested for loop ($j=1, j<=i, j++$)

Step IV: Display the number which the user had entered the same

Step V: The print the expression thus created on the next line

Step VI: Terminate the program

Conclusion:

Thus we have successfully executed the expression in Turbo C7 application.

Output: Enter the number of rows: 4

1
2 3
4 5 6
7 8 9 10

Practical 05

Write C program to find largest among number using array.

Algorithm:-

Start Turbo C application

Step I. Declare the variable and integer array ($a[10]$)

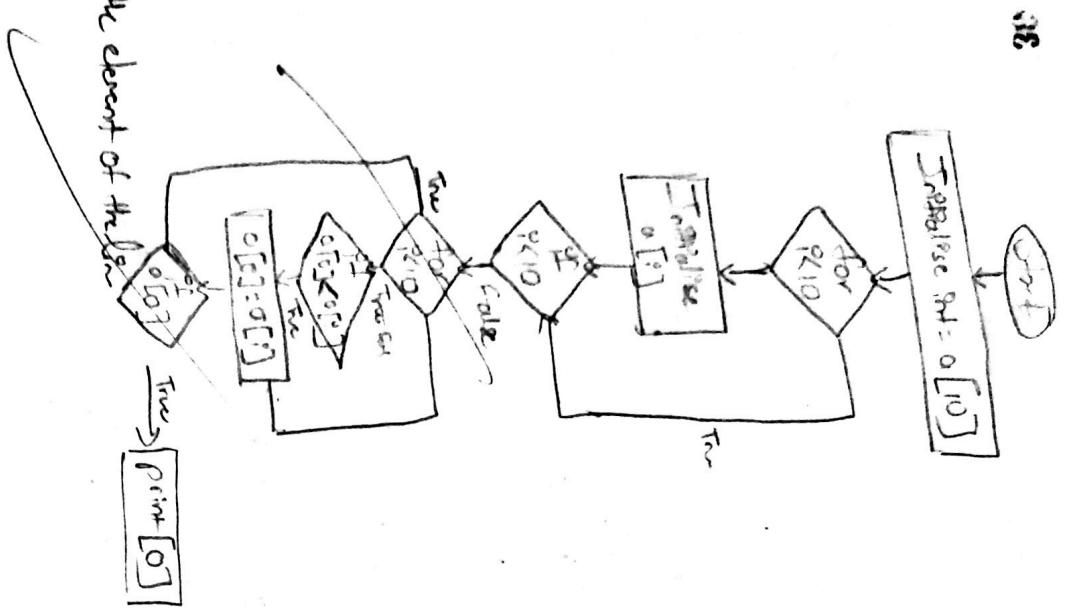
Step II. Enter the loop at $i=0$: $i < 10$ and ~~then~~ the value of $a[i]$ till $i < 10$. Exit for loop.

Step III. Enter the for loop at $i=0$, $i < 10$

~~if $a[0] < a[i]$
 $a[0] = a[i]$~~

Step IV. Run the above for loop for $i < 10$, exit the loop

Step V. Terminate the Program



100
72
18
6
5
4
12
97
17
23

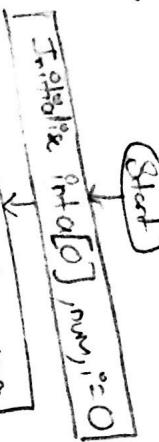
The largest no is 100

Source Code:

```

#include <conio.h>
#include <stdio.h>
void main()
{
    int a[10], i;
    clrscr();
    printf("Enter the element 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 no is %d", a[0]);
    getch();
}

```



Ques: Write a C Program to count even and odd number from the array.

Step I: Create an array , and take its size from user and define its element using loop

Step II: Display the size of array from the

Step III: Display the element of entered by user

Step I: Take the indicator in a for loop using which all the elements in the array exist

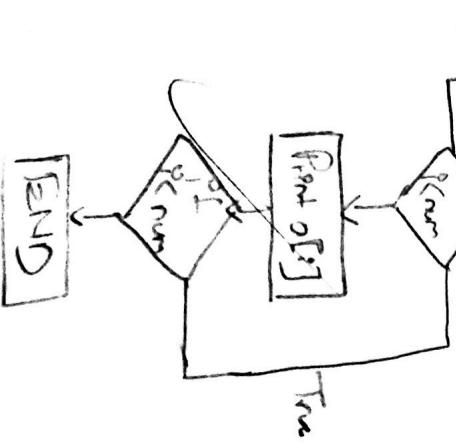
Step II: Display the even number from the array from for loop
(if $a[i] \% 2 == 0$)

Step III: Display even number from array

Step I: Display the odd number from array
(if $a[i] \% 2 != 0$)

Step II: Display odd number number from array

Step III: Terminate the program



```

Source Code:
#include <conio.h>
#include <stdio.h>
void main()
{
    int a[100], num, i;
    clrscr();
    printf("Enter the number of elements ");
    scanf("%d", &num);
    for(i=0, i<num, i++)
    {
        scanf("%d", &a[i]);
    }
    printf("Even no.:");
    for(i=0, i<num, i++)
    {
        if(a[i] % 2 == 0)
        {
            printf("\n%d", a[i]);
        }
    }
    for(i=0, i<num, i++)
    {
        if(a[i] % 2 != 0)
        {
            printf("\n%d", a[i]);
        }
    }
    getch();
}

```

Output:
Enter the no. of elements : 10
2 4 6 8 10
2 4 6 8
Odd no.: 3

Algorithms:

Start Turbo C++ application

~~Step 1: Initialize the int variable a [100], num ; and float variable sum = 0 and avg~~

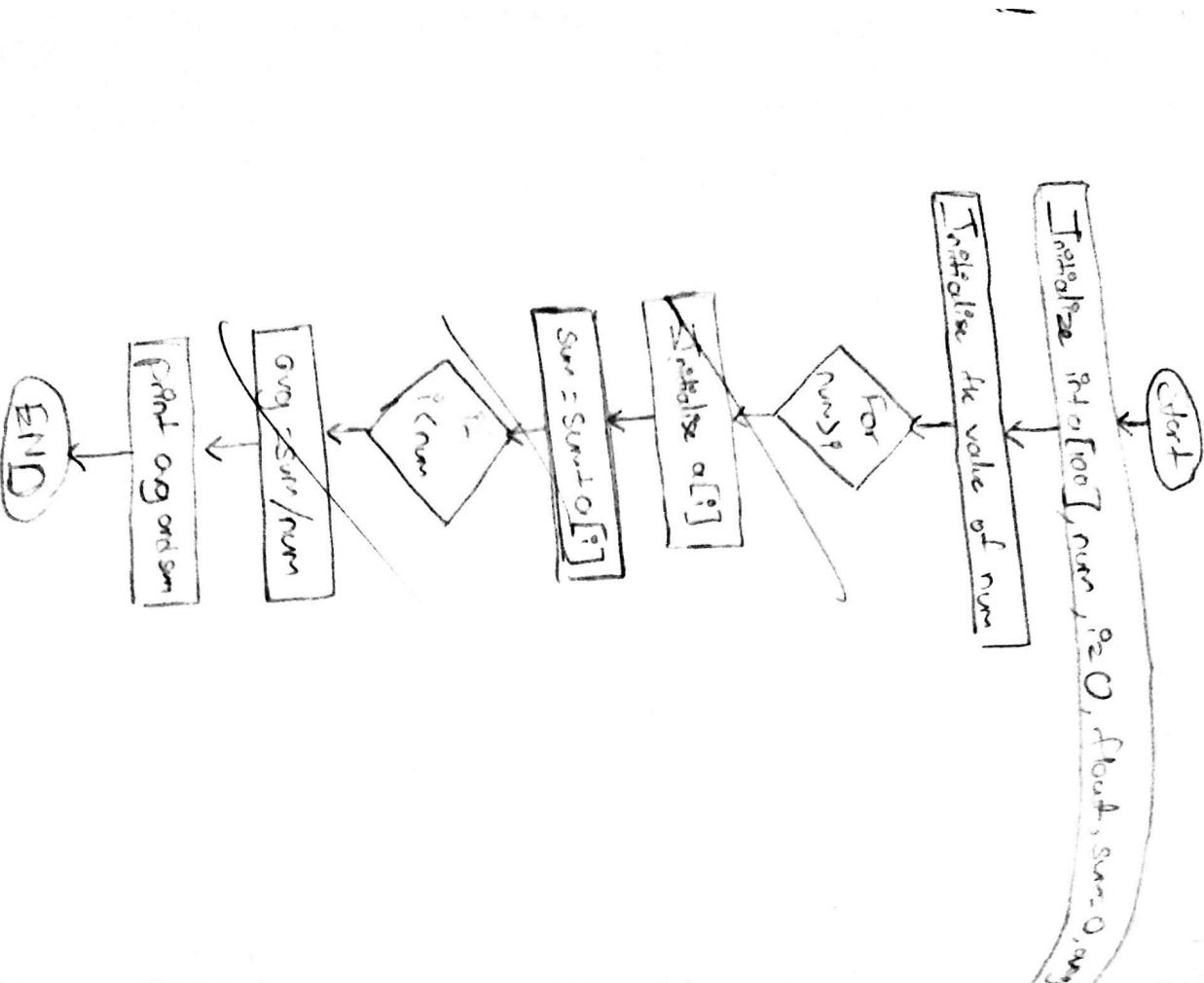
~~Step 2: Start the user for the length of array (<100)~~

~~Step 3: Start the user for the length of array (<100) and store the value in num using sum & avg~~

~~Step 4: Put avg = sum / num~~

~~Step 5: Print the value of sum & avg~~

~~Step 6: Terminate the program~~



Source Code:

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

```
{  
    int a[100], i, num;  
    float sum=0, avg;
```

```
clrscr();
```

```
printf("Enter no of element (<100)");
```

```
scanf("%d", &num);  
for (i=0, sum=0; i<num, i++)
```

```
{  
    printf("\n");
```

```
    scanf("%d", &a[i]);  
    sum += a[i];  
}
```

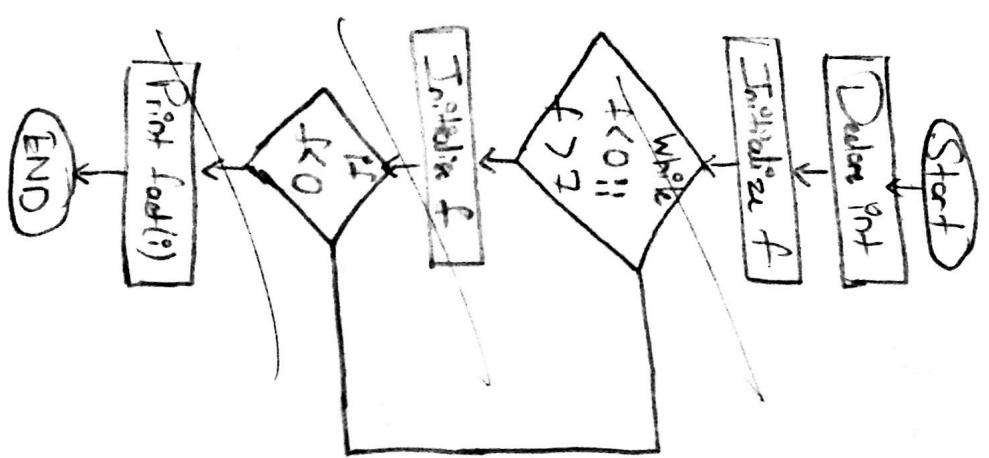
```
avg = sum/num;
```

```
printf("The sum of no. is %f and avg is
```

```
getch();  
}
```

Output:
Enter the no of elent 10
1
2
3
4
5
6
7
8
9
10
The sum of the number is 48.0000
and the avg is 4.80000

Conclusion: The program we have entered has been executed successfully.



I) Factorial of a number using recursion

Digitized by srujanika@gmail.com

~~Start the turbo C application~~

~~Step II~~ Declare function - prototype entering the main() function

~~Step 4~~ Declare a variable "f" in the main() function.

~~IV.~~ Use the whole function to ensure that entered value is in the range 0 to 7.

Step 3: Print the value entered by the function in Step 2.

^{3rd} Step. Terminate the program.

Source Code:

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

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

```
void main()
```

```
{
```

```
int f;
```

```
clrscr();
```

```
printf("Enter a number to find the factorial");
```

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

```
while (f<0 || f>7)
```

```
{
```

```
printf("Enter a number in range 0 to 7");
```

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

```
printf("Enter a number to find the factorial");
```

```
getch();
```

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

```
{
```

```
if (n==1)
```

```
{
```

```
return 1;
```

```
else
```

```
{
```

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

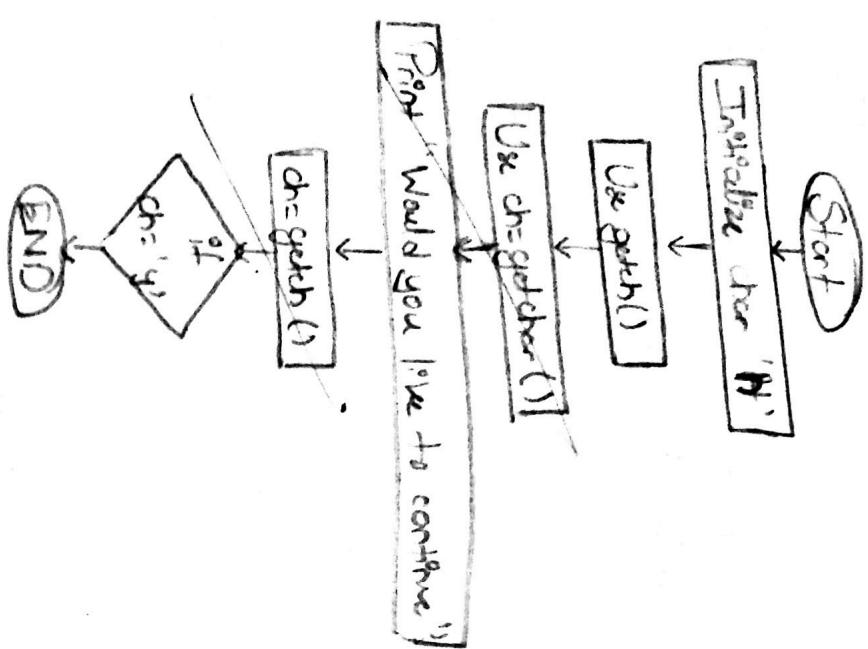
```
}
```

```
}
```

Output:-

Enter a number to find the factorial: 90
Enter a number in range 0 to 7: 6

720



W.A.P to show the use of `get()` function:-

Step 1: Start the turbo C application.

Step 2: Initialize a 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' on 'ch'.

Step 5: While `ch == 'y'`, keep accepting value for ch.

Step 6: Use the `getch()` method to store 'y' or 'n' in ch.

Step 7: Terminate the program

Source Code:

```
#include <stdio.h>
#include <conio.h>
Nord min()
{
    char ch: 'n'
    chscr();
    printf("Press any key to continue");
    getch();
    printf("Press any character");
    ch=getchar();
    Print ("Would you like to continue(y/n)?");
    ch=getchar();
    while(h=='y')
    {
        printf("Would you like to continue?");
        ch=getchar();
    }
}
```

Output:

Press any key to continue
 Enter any character: P
 Would you like to continue? y
 Would you like to continue? n

Start

Initialise ch 'A'

Use putch(ch)

Use putchar(ch)

END

W.A.P to show the use of put() function:

Step 1: Start the Turbo C application.

Step 2: Initialise a character 'ch' ~~on~~ to 'n'.

Step 3: Use the putch() and putchar() function with ch as the argument.

Step 4: - Terminate the program.

→ Source Code:

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

void main

{

```
char ch = 'A';
clrscr();
putch(ch);
//print("\n");
putchar(ch);
getch();
```

}

Output

A

A

→ ~~Conclusion~~ The tutorial on the getch and putch function has been done successfully

Practical 7

Ques: Write a C program to swap two number using pointer.

Algorithm:

Start Turbo C7 application

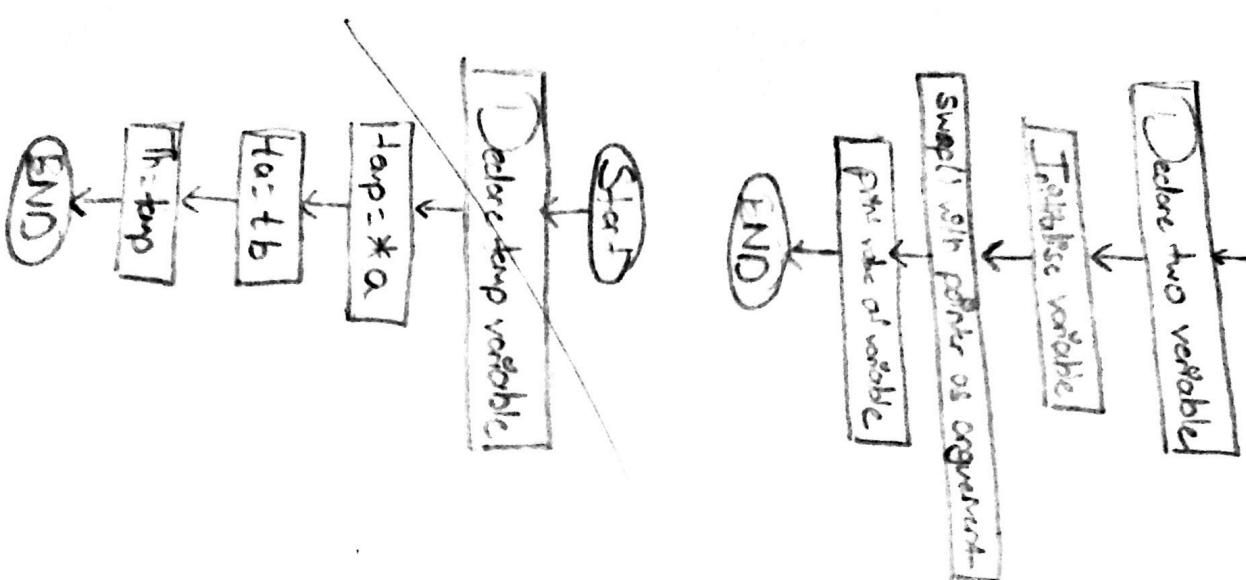
Step 1: Declare a function Prototype with two integer pointer as argument before entering main() function.

Step 2: Declare two variable and accept the value from user, Print the corresponding value using printf()

Step 3: Pass the address of the variable as argument from the function

Step 4: Print the respective value of the variable.

Note: Use the basis of swapping algorithm in the function definition but instead of normal variable use.



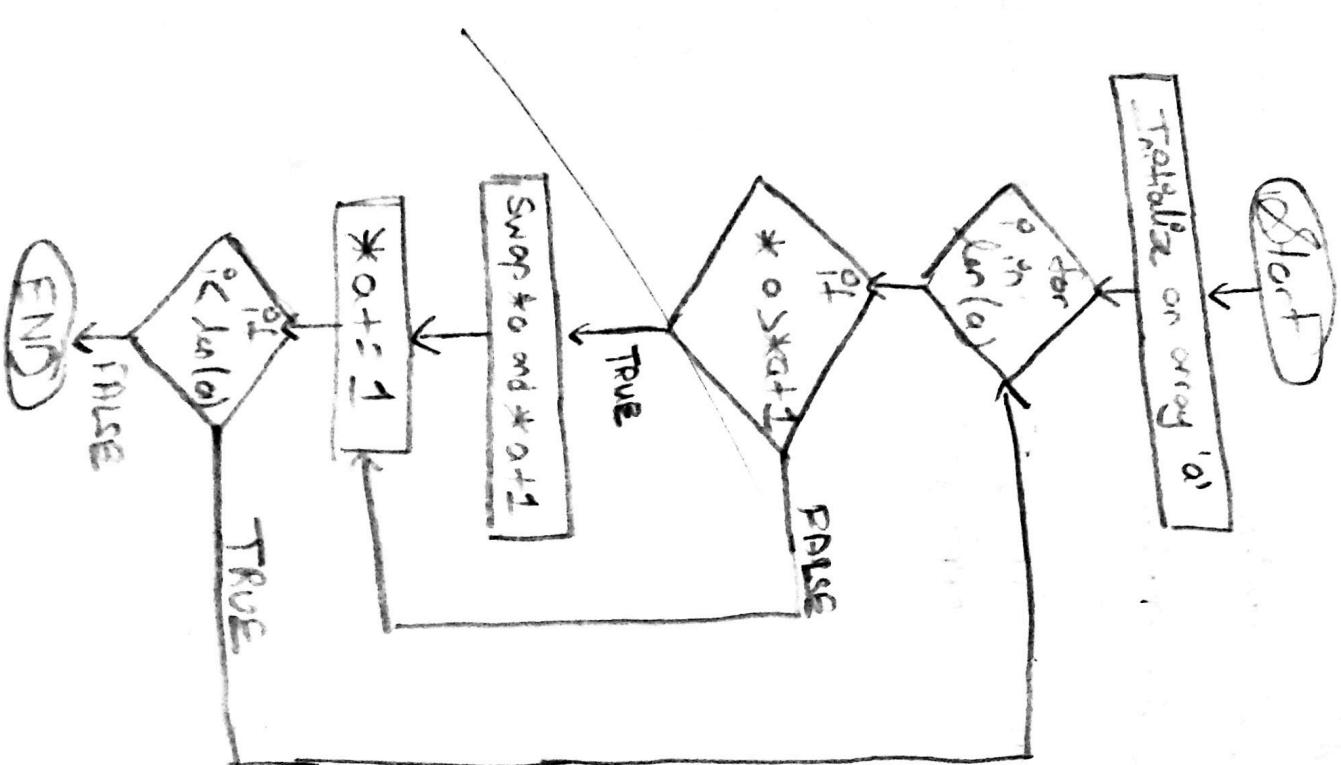
→ Source Code:

```
#include <stdio.h>
#include <conio.h>
void swap(int *a, int *b)
{
    int t;
    t = *a;
    *a = *b;
    *b = t;
}

main()
{
    int n1, n2;
    clrscr();
    printf("Enter the number\n");
    scanf("%d", &n1);
    printf("Enter the number\n");
    scanf("%d", &n2);
    printf("Before swapping n1:%d, n2:%d\n",
           n1, n2);
    swap(&n1, &n2);
    printf("After swapping n1:%d, n2:%d\n",
           n1, n2);
    getch();
}
```

Output:

Enter the number
77
Enter the number
88
Before swapping n1:77, n2:88
After swapping n1:88, n2:77



Ques: Sorting of an array using pointer.

→ Algorithm:

- 1: Initialize an array, i , j and temp variables.
- 2: Run a nested loop of $i = 0$ to $\text{len}(a)$ and j of $i+1$ to $\text{len}(a) - 1$
- 3: If $*a > *a+1$, swap the two values using basic swapping mechanism.
- 4: Print the swapped array
- 5: Terminate the program

→ Source Code

```

void sort(int a[], int *p)
{
    #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; j++)
            if (*a > *a + 1)
                {
                    temp = *a + 1;
                    *a + 1 = *a;
                    *a = temp;
                }
    }
    printf("%d is the sorted array", a);
}

```

Output:-

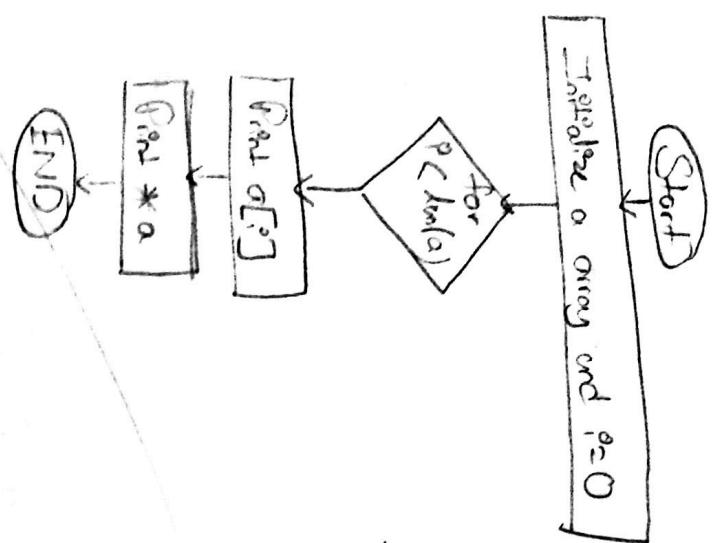
Insert elements into the array:

1
6
2
7

9
10

5
10

{1, 2, 5, 6, 7, 8, 9, 10, 10, 11} is the sorted array



III) Transversal of One-Dimensional Array using Pointer → Algorithm:

1. Start the turbo C7 application.
2. Initialize an integer array and a variable.
3. Run a for loop with $i=0$ to length of array
4. Print the data at the array and then use pointer to print the memory location
5. Terminate the program

→ Source Code:

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

```
void main()
```

```
{
```

```
int a[10];
clrscr();
printf("Enter the elements of the array :\n");
for(i=0;i<10;i++)
{
```

```
scanf("%d",&a[i]);
}
```

```
for(i=0;i<10;i++)
{
```

```
printf(" Data : %d ",*(a+i));
printf("Address:%u ",*(&a+i));
}
```

```
getch();
}
```

→ Output:
Enter the elements of the array

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

Data : 1
Address: 655506

Data : 2
Address: 655526

Data : 3
Address: 655546

Data : 4
Address: 655566

Data : 5
Address: 655586

Data : 6
Address: 6555a6

Data : 7
Address: 6555c6

Data : 8
Address: 6555e6

Data : 9
Address: 655606

Data : 10
Address: 655626

Data : 11
Address: 655646

Data : 12
Address: 655666

Data : 13
Address: 655686

Data : 14
Address: 6556a6

Data : 15
Address: 6556c6

Data : 16
Address: 655706

Data : 17
Address: 655726

Data : 18
Address: 655746

Data : 19
Address: 655766

Data : 20
Address: 655786

Data : 21
Address: 655806

Data : 22
Address: 655826

Data : 23
Address: 655846

Data : 24
Address: 655866

Data : 25
Address: 655886

Data : 26
Address: 655906

→ Conclusion: Thus, pointer have been used for various operations like showing memory location

Chmod