

PROGRAM

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int you no, mobile no,
    float percentage;
    char name [20];
    clrscr();
    printf ("Enter you no: ");
    scanf ("%d", &you no);
    printf ("Enter your percentage: ");
    scanf ("%f", &percentage);
    printf ("Enter your name: ");
    scanf ("%s", &name);
    printf ("Enter you mobile no: ");
    scanf ("%d", &mobile number);
    printf ("In you number is: %d\n", you no);
    printf ("In you percentage is: %f\n", percentage);
    printf ("In your name is: %s\n", name);
    printf ("In your mobile no is: %d\n", mobile number);
    getch();
}

```

Output: Enter you no number: 1234
Enter you percentage: 88
Enter you name: A Shlok
Enter you mobile no: 12345678910
You no is: 1234
You percentage is: 88
You name is: Shlok

PROJECT NO. 8

to display name, you no, percentage.

Aim: Write a C program to understand basic data type
2. input/output

Requirement: Turbo C.

Steps:

1. This C program creates a file and store info
2. we frequently use file for storing info which can be accessed by our programs
3. In order to store info permanently and retrieve it we need to use files and this program demonstrate file creation and writing data in that
4. The same code is used to create a file and store information
5. At last we compile the program, run it and the output is displayed

Conclusion: The given program gives us an idea about how built in datatype work in C and also about how user can give input and display output.

Result

ISSUE PRACTICAL NO: 2

Aim: Write a C program on operators and expressions

REQUIREMENTS: Turbo C

- Operators are the foundation of any programming language. Thus the functionality of C language is incomplete without the use of operators.
- Operators allow us to perform different kinds of operation on operands.

ARITHMETIC OPERATORS:

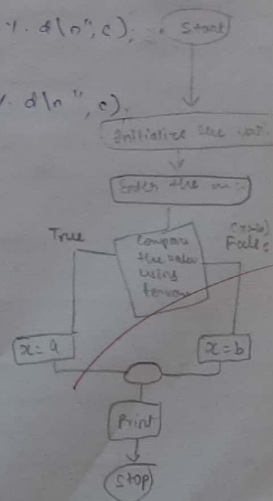
1. Addition: The '+' operator adds two operands (a+b).
2. Subtraction: The '-' operator subtracts two operands (a-b).
3. Multiplication: The '*' operator multiplies two operands (a*b).
4. Division: The '/' operator divides two operands (a/b).
5. Modulus: The '%' operator returns the remainder when first operand is divided by second (a%b).

PROGRAM CODE:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, c;
    clrscr();
    printf("Print two numbers");
    scanf("%d %d", &a, &b);
    c = a + b;
    printf("The sum of the nos is %d\n", c);
    c = a - b;
    printf("The diff of the nos is %d\n", c);
    c = a * b;
    printf("The mult of the nos is %d\n", c);
    c = a / b;
    printf("The div of the nos is %d\n", c);
    c = a % b;
    printf("The modulus of nos is %d\n", c);
    getch();
}
```

OUTPUT:

Print two no: 20, 70
 The sum of no: 30
 The diff of the no: is 10
 The mult of the no: : 200
 The div of the no: 2
 The modulus of the no: 0.



PROGRAM CODE:

```
#include <stdio.h>
int main ()
{
    int a,b,x;
    clrscr ();
    a=5;
    b=15;
    x = (a>b) ? a:b;
    printf ("%i,%i", x);
    getch ();
}
```

OUTPUT

15

ALGORITHM

STEP 1: Declare variables a,b,x as integers

STEP 2: Store the value of a as 5 and b as 15

STEP 3: Now to compare between a & b greater use ternary operator as to find

STEP 4: Use printf function to display output

CONCLUSION:

These programs help us in having better understanding about operators & expression

Practical No: 3

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

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

Algorithm

STEP 1: Declare a variable as integer and assign its value 20

STEP 2: Now to compare whether 20 is greater than 15 use if statement.

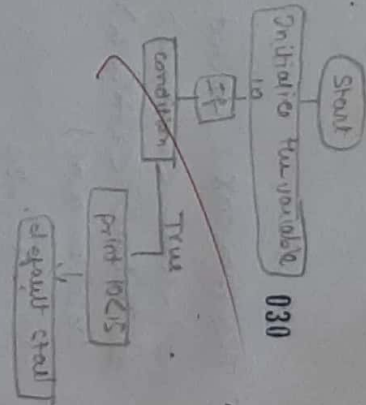
STEP 3: If the condition is true, print '20 is less than 15 and if condition is false skip the if statement & print I am not in if

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

STEP 1: Declare a variable as integer & assign its value as 20

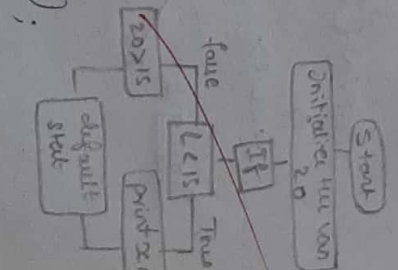
STEP 2: Now to compare the given value if it is greater or not use if else cond. stat.

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



OUTPUT:
I am not in if.

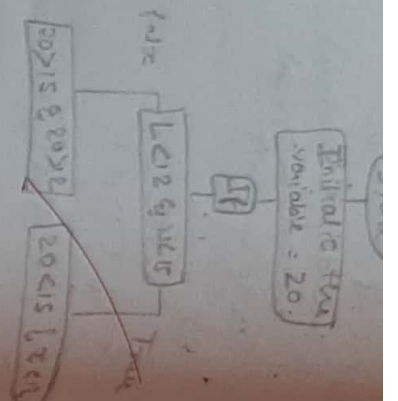
b) CODE : If else statement
#include <stdio.h>
#include <conio.h>
int i=20;
if (i<15)
{
printf ("20 is greater than 15\n");
}
else
{
printf ("20 is greater than 15\n");
}
getch();



OUTPUT:
20 is greater than 15

c) Program 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 & 12\n");
        }
        else
        {
            printf("20 is greater than 15 & 12\n");
        }
    }
    getch();
}
```



Output:

20 is greater than 15 & 12

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

Algorithm:

STEP 1: Declare a variable on integer and assign value 20

STEP 2: Now we nested if logic to compare if given number is greater or not

STEP 3: If condition is true then go to 2nd condition if second cond. is also true then print that 20 is greater than 15 & 12. If any of condition are not true then skip the part and print 20 is greater than 15 & 12

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

Support:

Q8.

PROBLEM NO: 4.

AIM: To display the prime numbers using the loop

ALGORITHM:

STEP 1: Initialize the variable and a count variable loop variable and one as count variable

STEP 2: Initialize a for loop. 1 to 50 let the count variable to be zero.

STEP 3: Next another loop within the loop in step 2 that goes to 2 to the first variable $i/2$

STEP 4: Use the if and statement to check whether first loop variable i . 2nd loop variable $i/2$, if two increment count variable by 1

STEP 5: Come out of the 2nd loop & check whether the count variable is 0 if you print the no

STEP 6: Terminate the program

CONCLUSION: We have successfully executed a program of prime no. using for loop

CODE:

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

```
{
    int n,i,j;
    clrscr();
    printf("The Prime numbers are:");
```

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

```
{
    a=0;
    for (j=2; j<=i/2; j++)
```

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

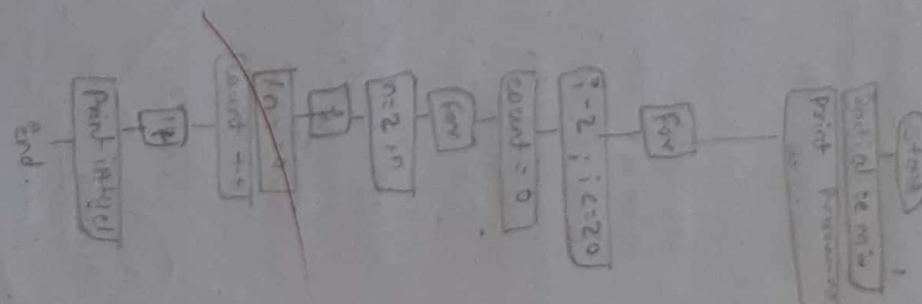
```
{
    a++;
```

```
}
if (a==0)
```

```
{
    printf("%d / n", i);
```

```
}
getch();
}
```

032



569

#include <stdio.h>

#include <conio.h>

void main ()

{ int n1=0, n2=1, n3, i, number;

clear();

printf ("Enter number of element 'n'");

scanf ("%d", &number);

printf ("%d", n1, n2);

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

{

n3 = n1 + n2;

printf ("%d", n3);

n1 = n2;

n2 = n3;

}

getch();

}

Output:

Enter number of Element (0

0 11235813213455

033

7) Write C program on fibonacci series.

Algorithm:STEP 1: Start TurboCSTEP 2: Declare the variable n1, n2, n3, numberSTEP 3: Initialize the variable n1=0, n2=1, n=0STEP 4: Enter the no. of term of fibonacci series to be printedSTEP 5: Print first 2 terms of series as n1=0 & n2=1STEP 6: Use the for loop as per below step~~n3 = n1 + n2~~~~n1 = n2~~~~n2 = n3~~

Increase the value of i increment each time by 1

STEP 7: Print the value of numberSTEP 8: EndConclusion: Thus we have successfully executed fibonacci series on TurboC.

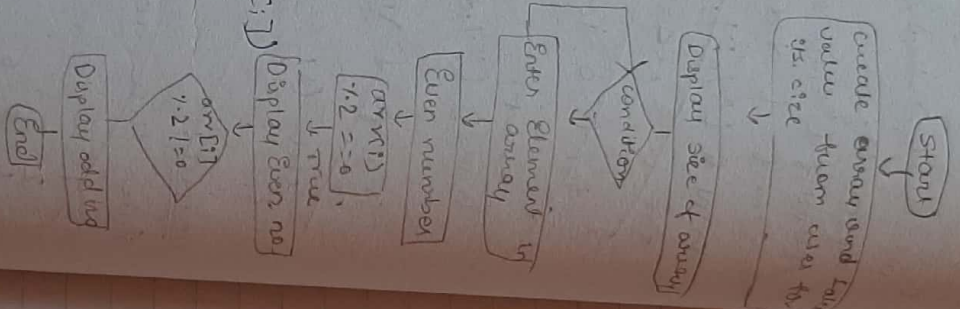

```
#include <conio.h>
#include <stdio.h>

void main()
{
    int a[10], i;
    clrscr();

    printf("Enter the element of the list");
    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();
}
```



Practical No: 5.

Aim: C program to find largest array number using array

Algorithm:

STEP 1: ~~Start~~ ~~hadoop~~ application

STEP 2: Declare the variable i and integer array a[10]

STEP 3: Enter the for loop at i=0, i<10 and 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 conditional statement to decide 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

ii) WAP to find the count the even and odd no. in the array

Ans: WAP to print the no. odd and even no. in the array

ALGORITHM:

STEP 1: Locate an array stating size from user & display it as element using loop

STEP 2: Display the size of array.

STEP 3: Display the element of array entered by user

STEP 4: Take the iterator in a for loop using array all the elements in the array exist

STEP 5: Display even no. in the array from for loop

STEP 6: If (array[i] % 2 == 0)
Display even no. from given array

STEP 7: Display the odd no.
if (array[i] % 2 != 0)
then display odd no.

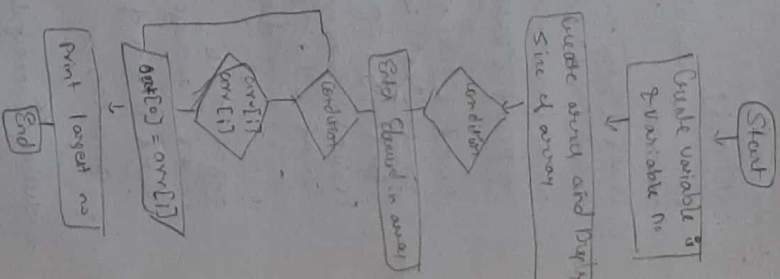
Code:

OUTPUT:

Enter the element of the list

12
23
2
2
12
55
3
1
22
100

Give largest no. is 100.




```

CODE:
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, i;
    float num[100], sum, avg;

```

```

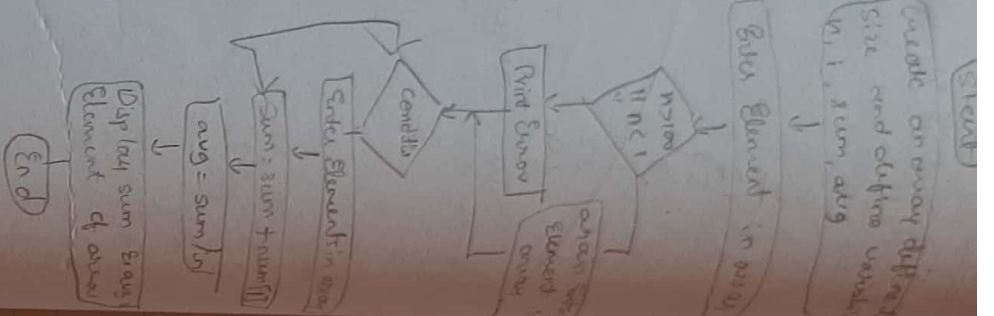
    clrscr();
    printf("Enter the no. of elements");
    scanf("%d", &n);
    while (n > 100 || n < 1)
    {
        printf("Error!");
        printf("Enter no. again");
        scanf("%d", &n);
        for (i = 0; i < n; i++)
        {
            printf("Enter no. : ");
            scanf("%d", &num[i]);
            sum = sum + num[i];
        }
        avg = sum/n;
        printf("Average of array is %f", avg);
        printf("Sum of array is %f", sum);
        getch();
    }
}

```

Conclusion: Successfully executed the program

OUTPUT:

Enter no. of elements: 5
 Enter no. of element: 5
 Enter no: 1
 Enter no: 2
 Enter no: 3
 Enter no: 4
 Enter no: 5
 Average of array is 3.000
 Sum of array is 15.000



Write a program to find sum & average of element in array

Algorithm

- STEP 1: Create an array, take its size from user & define its element in for loop
- STEP 2: Now check if size is (n > 100 || n < 10) then print Error
- STEP 3: Now enter element in array using get for loop & find the sum i.e. sum = sum + num[i]
- STEP 4: Find average by avg = sum/n
- STEP 5: Display average & sum & element in array

Terminate program

Conclusion: Successfully executed the program

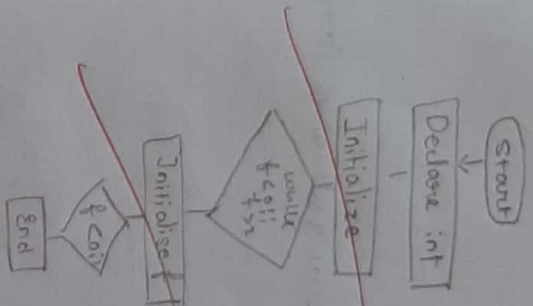
Handwritten signature

PRACTICAL NO: 6.AIM: Factorial of a no. using recursionALGORITHM:

A)

STEP 1: Start the turbo C applicationSTEP 2: Declare the func prototype Entering the main() functionSTEP 3: Define a variable 'f' in the main() funcSTEP 4: Use the while func to ensure that entered value is in range 0 to 17STEP 5: Print the value Entered by the func in stepSTEP 6: Terminate the prog.

Flowchart



CODE: A
#include <stdio.h>
#include <conio.h>
void main()

{ int b;
clrscr();
printf ("Enter the no. to find the factorial of (0 to 7):");
scanf ("%d", &b);
while (b < 0 || b > 7)

{ printf ("Invalid", b);
scanf ("%d", &b);

} printf ("%d", fact(b));
getch();

} int fact (int n)

{ if (n > 1)

{ return n * fact(n-1);

else

{ return 1;

}

Output:

Enter the no. to find the factorial of (0 to 7): 6:
720

B] With a program which shows the use of getch() func.

STEP 1: Start the Turbo C app.

STEP 2: Initialize the character variable.

STEP 3: Use the getch() method to accept the char.

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

STEP 5: While char 'y' keep accepting value for ch.

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

STEP 7: Terminate.

Conclusion: The program was successfully executed.

CODE B:

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

```
{
    clrscr();
    clrscr();
    printf("Press any key to continue");
```

```
getch();
```

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

```
printf("would you like to continue y/n");
```

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

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

```
{
```

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

```
    getch();
```

```
}
```

OUTPUT:

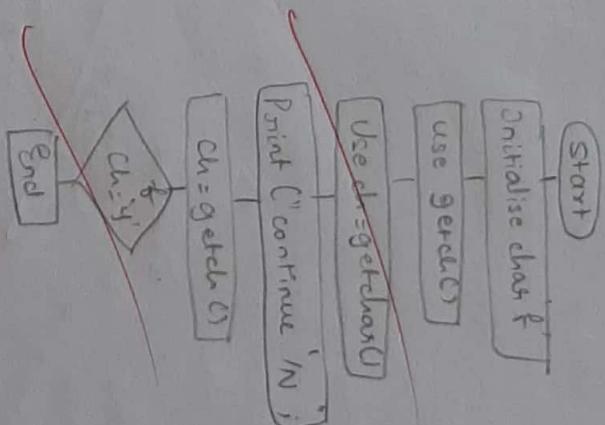
Press any key to continue

Enter any character: A

would you like to continue? y/n

would you like to continue? y/n

3. Flowchart



Q10

CODE C:

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

void main()

{

clrscr();

char ch;

printf("n");

printf("n");

printf("n");

printf("n");

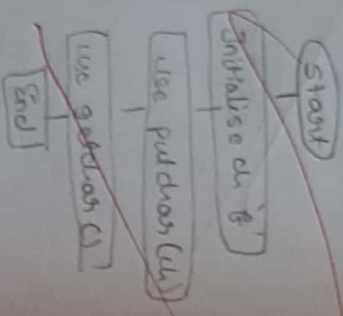
}

OUTPUT:

A.

n.

Flowchart



041

C) we do know the use of printf function.

STEP 1: Start the main C app

STEP 2: Initialize the character 'n' to 'n'.

STEP 3: Use the putchar() and putchar() function with 'n' as the argument.

STEP 4: Terminate the program.

CONCLUSION:

The function of getch() and printf() have been used in the program successfully.

Remark:

PRACTICAL NO: 7

Aim: Write a program to find the swapping of 2 no.
(pointer)

ALGORITHM:

STEP 1: Start the turbo C application

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

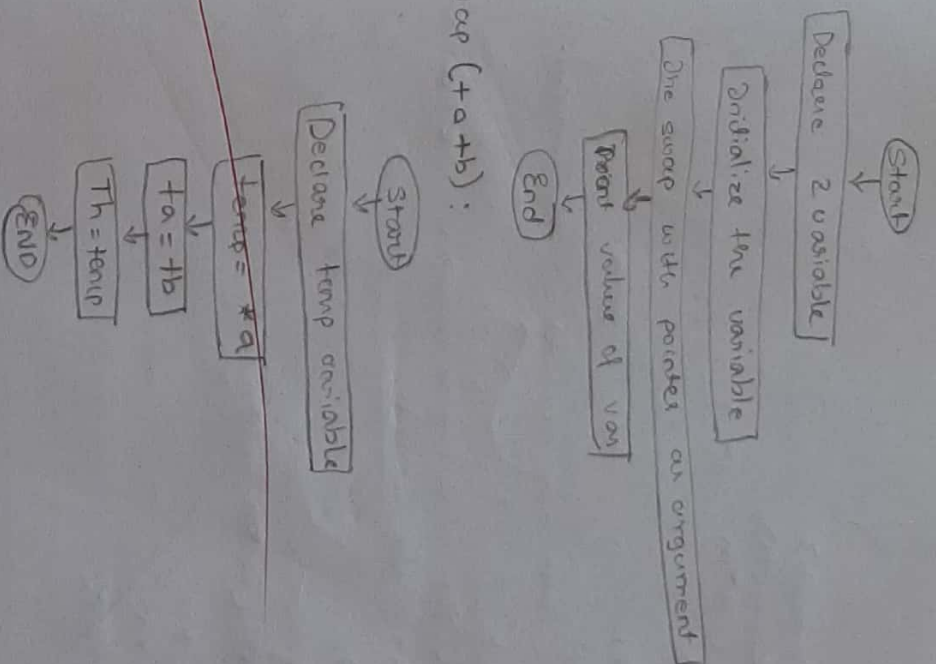
STEP 3: Declare 2 variable & accept their value from user and the separate value using printf()

STEP 4: Pass the address of the variable as argument for the

STEP 5: ~~Print the reciprocal value of variable~~

STEP 6: Use the basic swapping algo in the function but instead of normal variable use

Swap (+a +b):



CODE:

#include <stdio.h>

#include <conio.h>

void swap (int *m, int *n);

void main()

{ int x, y;

obsch:

printf ("Enter two numbers to be swapped:");

scanf ("%d %d", &x, &y);

printf ("The values before swapping are %d and %d respectively x, y);

getch();

}
void swap (int *m, int *n)

{ temp = *m;

*m = *n;

*n = temp;

}

OUTPUT:

Enter the 2 no. to be swapped: 12

24

The no. before swapping are 12 8 24
The no. after swapping are 24 8 12

2] Sorting of array using pointers

ALGORITHM:

STEP 1: Initialize an integer array, i, j and temp variable

STEP 2: Run a nested loop of i = 0 to len-1 and of j = 0 to len(a)-i.

STEP 3: if a[i] > a[j], swap the two values using basic swapping logic

STEP 4: Print the swapped array

STEP 5: Terminate the program

5] Write a program to find one dimensional array using pointer

Algorithm:

STEP 1: Start the Turbo C application

STEP 2: Declare an integer array & a variable

STEP 3: Run a for loop with $i=0$ to length of array

STEP 4: Print the done of array & then use pointer to point the memory loc.

STEP 5: Terminate the program

2] 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; j++)
```

```
        {
```

```
            if (a[i] > a[j])
```

```
            {
```

```
                temp = a[i];
```

```
                a[i] = a[j];
```

```
                a[j] = temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("Sorted array:");
```

```
    getch();
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

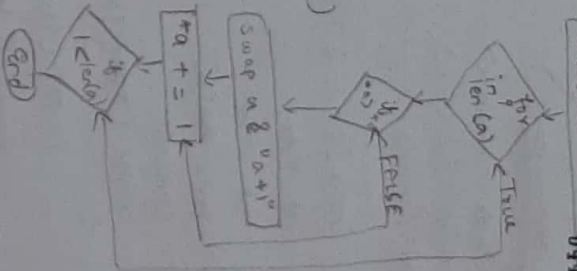
```
}
```

```
}
```

Sort

Initialize an array 'a'

044



Output:

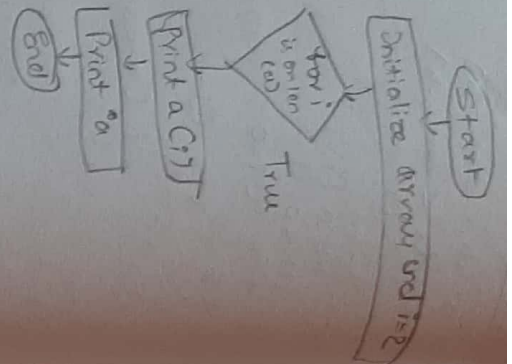
Enter elements into the array

1 2 3 4 5 6 7

1, 2, 3, 4, 5, 6, 7 in the sorted array

3 # Source code:

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



OUTPUT:

the address of a[0] = 65516
 the value of a[0] = 7
 the address of a[1] = 65518
 the value of a[1] = 5
 the address of a[2] = 65520
 the value of a[2] = 9
 the address of a[3] = 65522
 the value of a[3] = 4
 the address of a[4] = 65524
 the value of a[4] = 2

CONCLUSION:

The program to find one dimensional array using pointer is done subsequently.

Amogh