

### Practical No 1

33

Ans: write a c program to understand the basic datatype and I/O:

Theory:  
write a c program to display student name, rollno, mobile no and percentage.

#### ALGORITHM:

Step 1: Declare a variable name, rollno as integer,  
also declare name, mobile no as characters  
and 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  
and store in its allocated memory

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

```
SS  
Code:  
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    int rollno;  
    char name[20], mobile-no[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", &mobile-no);  
    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", mobile-no);  
    printf("Student's percentage: %f\n", percentage);  
    getch();
```

CONCLUSION:

The given program gives us an idea about how built-in datatypes work in c and also about how user can give input and display output.

Name:

Manisha  
1899  
6921391182  
76.000

OUTPUT:

Entee student's name:  
Manisha

Entee student's rollno:  
1899

Entee student's mobile no:  
6921391182

Entee student's percentage:  
76

Student's name: Manisha.

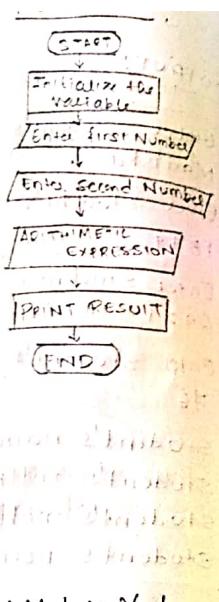
Student's rollno: 1899.

Student's mobile no: 6921391182

Student's percentage: 76.000

18  
CODE:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, add, sub, mult, div;
    clrscr();
    printf("Enter first number:");
    scanf("%d", &a);
    printf("Enter second number:");
    scanf("%d", &b);
    add = a + b;
    sub = a - b;
    mult = a * b;
    div = a / b;
    printf("the addition of %d and %d is %d,\n", a, b, add);
    printf("the subtraction of %d and %d is %d,\n", a, b, sub);
    printf("the multiplication of %d and %d is %d,\n", a, b, mult);
    printf("the division of %d and %d is %d,\n", a, b, div);
    getch();
}
```



### PRACTICAL NO. 2.

35

AIM: Write a C program on operator and expression.

THEORY: Write a program to create a dynamic calculator

#### ALGORITHM:

Step 1: Declare a variable name for first and second number as integer

Step 2: Now use Scan function to receive input from user

Step 3: Now to add two numbers given by user, use the expression  $a+b$ .

Step 4: Now to subtract two number given by user, use the expression  $a-b$ .

Step 5: Again use expression  $a*b$  if user wishes to multiply the two inputs.

Step 6: use expression  $a/b$  if user wishes to divide the two inputs.

Step 7: Now use printf function to display output.

b] write a program in c to explain ternary operator.

Step 1: Declare Variables a,b and x as integers.

Step 2: Store the value of a as 5 and 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 to in having better understanding about operators and expressions.

Normal

#### OUTPUT:

```
Enter first number 3
Enter second number 2
The addition of 3 and 2 is 5.
The subtraction of 3 and 2 is 1
The multiplication of 3 and 2 is 6
The division of 3 and 2 is 1
```

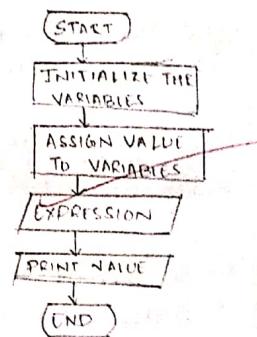
#### b] CODE:

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

#### OUTPUT

15

#### FLOWCHART



## 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:

(a) Step 1: Declare a variable as integer and assign its value be 20.

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

Step 3: If the condition is true print then 20 is less than 15 and if condition is false skip the if statement and print I am not in if

write a program in c to explain if-else statement

Step 1: Declare a Variable as Integer and assign its Value be 20.

Step 2: Now to Compare the given Value if its greater or not use if-else Conditional Statement

(a) code:  
// IF STATEMENT  
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
 int i = 10;  
 clrscr();  
 if (i > 15)  
 {  
 printf("i is less than 15\n");  
 }  
 else  
 printf("i am not in if\n");  
 getch();  
}

(b) OUTPUT:  
i am not in if.

### (b) CODE:

// IF ELSE STATEMENT  
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
 int i = 20;  
 clrscr();  
 if (i < 15)  
 {  
 printf("i is smaller than 15\n");  
 }  
}



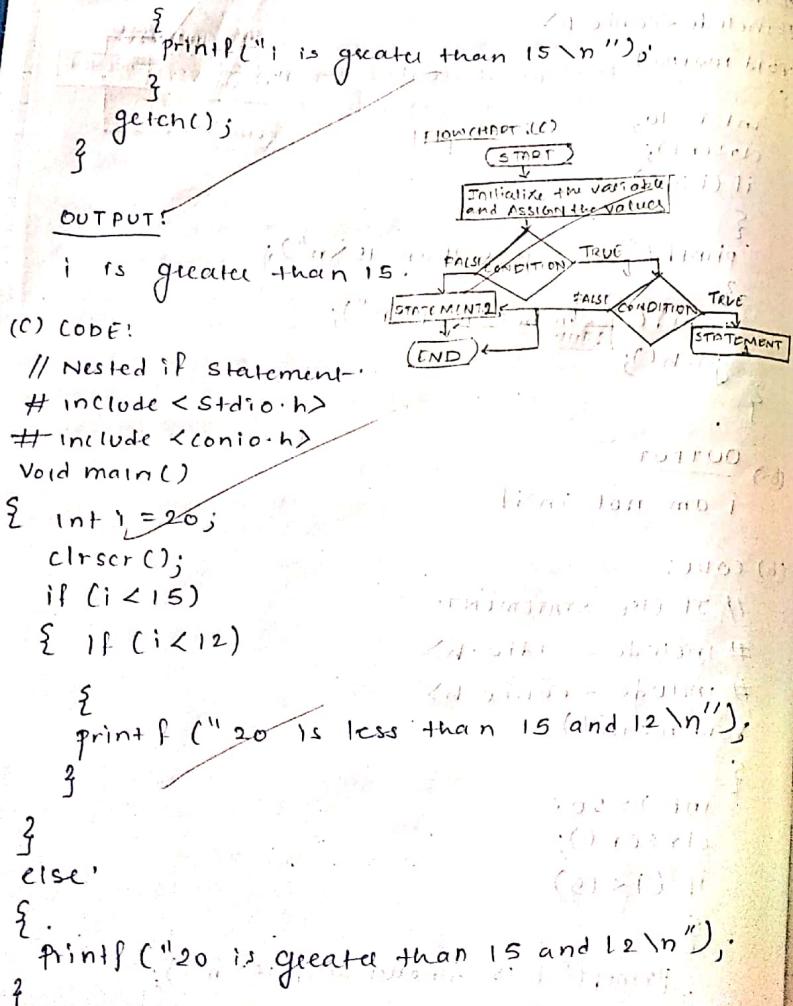
Step 3: If condition is true then print 20 is less than 15 or if condition is false then print 20 is greater than 15.

c) write a program in c to explain nested if statement.

Step 1: Declare a variable as integer and assign value is 20.

Step 2: Now use nested if logic to compare if given no 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 and 12. If one of the condition are not true then skip the part and print 20 is greater than 15 and 12.



188

### CONCLUSION!

These programs help us to understand the working of if, if-else, and nested if conditional statements.

Journal

getch();

}

OUTPUT:

20 is greater than 15 and 12

## PRACTICAL NO. 4

AIM: To display prime numbers using for loop.

ALGORITHM..

Step 1: Initialise three variables out of which two are loop variable and one is a count variable.

Step 2: Initialise a for loop within the loop from 2 to 20 value. Set the count variable to 0 zero.

Step 3: put another loop within the loop in step 2. that goes from 2 to the first loop-variable / 2.

Step 4: Use the if-conditional statement to check whether ( $1^{\text{st}}$  loop variable  $i$ ,  $2^{\text{nd}}$  loop variable  $\leq 0$ ). If true, increment count variable by 1

Step 5: Come out of the second loop and check whether the count variable is 0. If true, print the number ( $1^{\text{st}}$  loop variable)

Step 6: Terminate the program

CONCLUSION: prime numbers were displayed using for loop.

```
# CODE:  
# include <stdio.h>  
# include <conio.h>  
Void main()
```

```
{ int n, i, a;  
clrscr();  
printf("The prime numbers are: ");  
for (i = 2, a = 0; 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
```

1233 38  
1233 42  
1233 46

1233 48

1233 50

1233 52

1233 54

1233 56

1233 58

1233 60

1233 62

1233 64

1233 66

1233 68

1233 70

1233 72

1233 74

1233 76

1233 78

1233 80

1233 82

1233 84

1233 86

1233 88

1233 90

1233 92

1233 94

1233 96

1233 98

1233 100

1233 102

1233 104

1233 106

1233 108

1233 110

1233 112

1233 114

1233 116

1233 118

1233 120

1233 122

1233 124

1233 126

1233 128

1233 130

1233 132

1233 134

1233 136

1233 138

1233 140

1233 142

1233 144

1233 146

1233 148

1233 150

1233 152

1233 154

1233 156

1233 158

1233 160

1233 162

1233 164

1233 166

1233 168

1233 170

1233 172

1233 174

1233 176

1233 178

1233 180

1233 182

1233 184

1233 186

1233 188

1233 190

1233 192

1233 194

1233 196

1233 198

1233 200

1233 202

1233 204

1233 206

1233 208

1233 210

1233 212

1233 214

1233 216

1233 218



```

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

```

~~OUTPUT:~~

0 Enter a no of element: #10

1

2

3

5

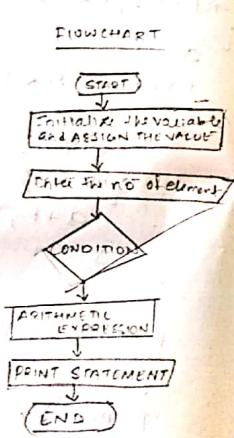
8

13

21

34

55



(b) write a c program on fibonacci series.

ALGORITHM:

STEP 1: Start -> the Turbo C

STEP 2: Declare Variables n1, n2, n3, i, number

STEP3: Initialize the variables n1=0, n2=1, number=0

STEP4: Enter the number of terms of Fibonacci series,  
to be printed

STEPS: print first term of series as n1=0, n2=1

STEP6: use the for loop as per following step

$$n3 = n1 + n2$$

$$n1 = n2$$

$$n2 = n3$$

increase the value of i element at each time  
by 1.

STEP7: print the value of number

STEP 8: End.

CONCLUSION: Thus, we have successfully execute fibonacci  
series on turbo C.

(c) AIM: write a c 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: Declare the variable rows, i, j, number=1.

Step 3: Display the number rows

Step 4: Enter the for loop as  $i = 1, i \leq \text{rows}, i++$ .

Step 5: Now create nested for loop as  $j = 1, j \leq i, ++j$ .

Step 6: Display the number as per usce enter the sequence from  $i = 1$ .

Step 7: Increment the number from 1.

Step 8: Display the space

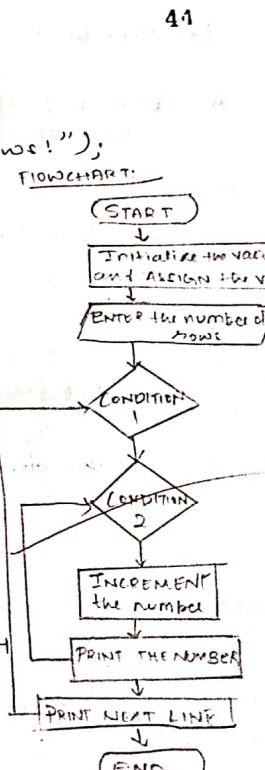
Step 9: End.

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

#### OUTPUT:

Enter the number of rows: 4

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



**CONCLUSION:** Thus, we have successfully execute the given expression on turbo C using nested for loop.

Kimball

## PRACTICAL NO. 5

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

Algorithm:

Step 1: Start turbo C

Step 2: Declare two variables i and integer in 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 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

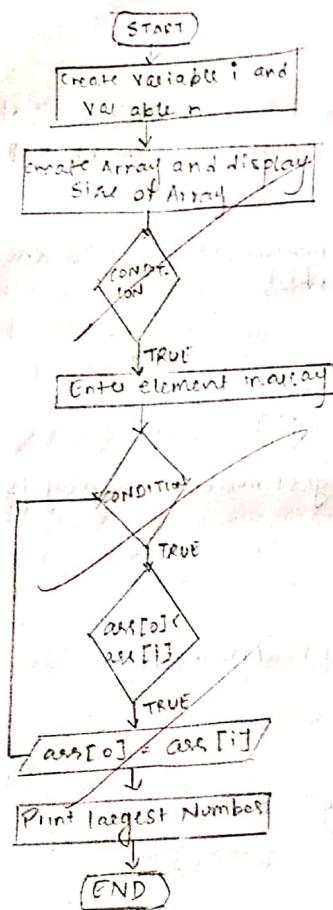
Conclusion: Thus, we have successfully executed the program.

46

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, n;
    float arr[100];
    clrscr();
    printf("Enter the number of elements from (1 to 100): ");
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        printf("\nEnter a number: ", i+1);
        scanf("%f", &arr[i]);
    }
    for(i=1; i<n; i++)
    {
        if(arr[0] < arr[i])
            arr[0] = arr[i];
    }
    printf("\nLargest number in array is %f", arr[0]);
    getch();
}
```

OUTPUT:

```
Enter the number of element from (1 to 100): 3
enter a number: 33
enter a number: 67
enter a number: 86
largest number in array is 86.00
```



Aim: write a c program to print a number of odd and even number in array.

Algorithm:

~~Step1: Create an array. Take its size from user and define its element using loop.~~

~~Step2: display the size of the array~~

~~Step3: Display the element of array entered by user.~~

~~Step4: Take the initiator in for loop in which all the element in array exists.~~

~~Step5: Display even number from array using for loop.~~

~~Step6: if (arr[i] % 2 == 0) then  
display the even number in given array.~~

~~Step6: Display odd number from array using for loop  
if (arr[i] % 2 != 0) then display the Odd  
number in the given array~~

~~Step7: Terminate the program.~~

CONCLUSION: Thus, we have successfully executed the program.

```

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

```

OUTPUT:  
Enter the size of array: 3

78

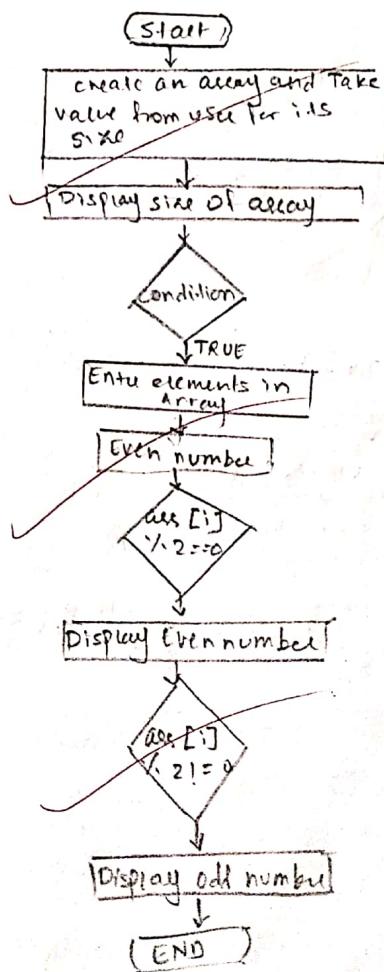
89

34

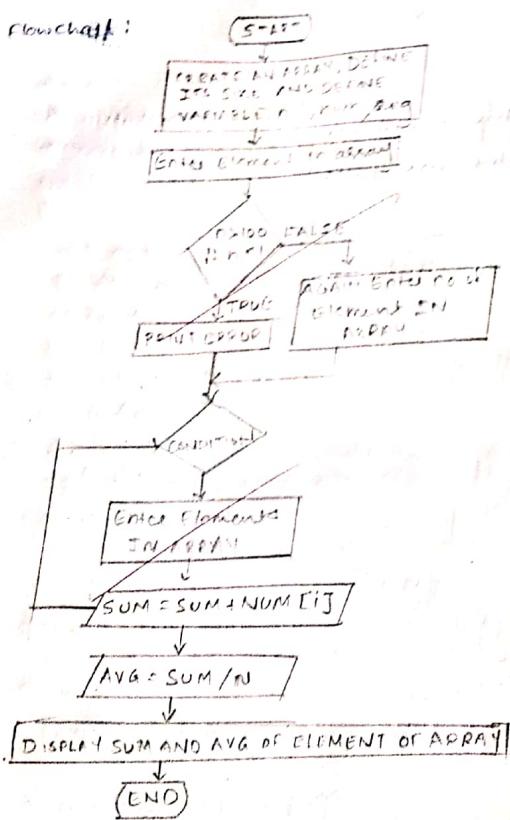
Even no in the array are! 78 34

Odd no in the array are! 89

Flowchart:



rough work:



49

Aim: write a c program to find the sum or average of element in array.

Algorithm:

Step 1: Create an array Take its size from array user and define its element using while loop & declare the integer in array after

Step 2: Now check if size is 10 for loop at  $i=0$  till  $i < 10$  and we the value of  $a[i]$  till  $i < 10$ . Exit this for loop and print the result

Step 3: Now enter elements in array using for loop and find the sum ie  $sum = sum + num[i]$

Step 4: find average by  $avg = sum/n$

Step 5: Display average and sum of the element in array

Step 6: Terminate the program.

Conclusion: Successfully executed the program to find the sum or average of element in array

Method:

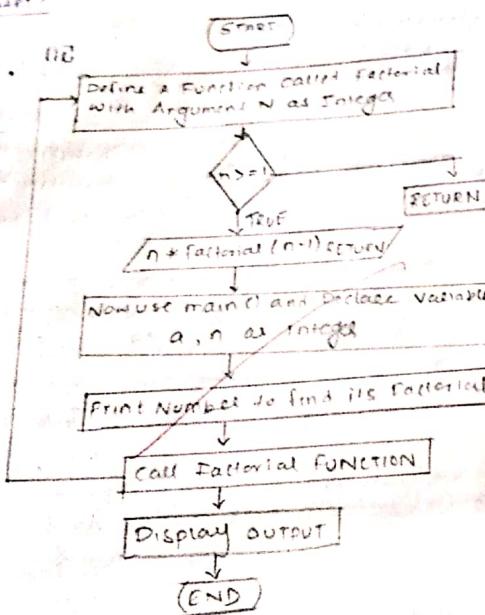
```

#include <stdio.h>
#include <conio.h>
Void main()
{
    int n;
    float num[100], sum, avg;
    clrscr();
    printf("enter the no. of element:");
    scanf("%d", &n);
    while (n > 100 || n < 1)
    {
        printf("error! or number should be in range of
               1 to 100 \n");
        printf("enter the number again:");
        scanf("%d", &n);
        for (i = 0; i < n; i++)
        {
            printf("Y.d enter no: ", i + 1);
            Scanf("Y.f", &num[i]);
            sum = sum + num[i];
        }
        avg = sum / n;
        printf("average = %2f", avg);
        printf("\n sum = %2f", sum);
        getch();
    }
}


```

~~#OUTPUT:~~

Enter the no. of element: 3  
 enter the number again: 3  
 1 enter no: 3  
 2 enter no: 4  
 3 enter no: 5  
 average = 4.000  
 sum = 12.000

**FLOWCHART****PRACTICAL NOTE**

51

**AIM:** Programs on Functions:

Q. Write a C program to find factorial of a number using recursion:

Algorithm:

Step 1: Define a function called factorial with argument n as integers

Step 2: In this function use if conditional statement to check whether the number is greater than 1 then return  $n * \text{factorial}(n-1)$  else return 1

Step 3: Now use main() then declare a variables a, n as integers.

Step 4: Now enter number to finds its factorial.

Step 5: Now call the function factorial and display the answer.

Step 6: Terminate the program.

-18

CODE:

```
#include <stdio.h>
#include <conio.h>
int Factorial (int n)
{
    if (n >= 1)
        return n * Factorial (n - 1);
    else
        return 1;
}
Void main()
{
    int a, n;
    printf ("Enter a number to find its factorial:");
    Scanf ("%d", &n);
    a = Factorial (n);
    printf ("Factorial of %d is %d", n, a);
    getch ();
}
```

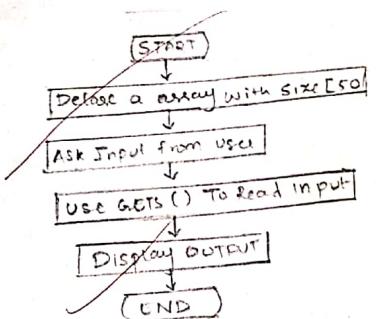
~~CONCLUSION: Successfully executed the program to find factorial using recursion.~~

OUTPUT:

Enter a number to find its factorial: 5  
Factorial of 5 is 120

#OUTPUT:  
Enter your name: Manisha  
your name is Manisha

FLOWCHART:



53

Q. Write a C program which shows the use of gets() function

Algorithm:

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

Step 2: Now ask your name by user.

Step 3: Now use gets() to read the input from user.

Step 4: Print your name on screen.

Step 5: Terminate the program.

#CODE:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
Void main()
{
    char name[50];
    printf("Enter your name : ", gets(name));
    printf("\n Your name is %s ", name);
    getch();
}
```

CONCLUSION: Successfully executed the program

88

Q8. write a c program 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.

#CODE:

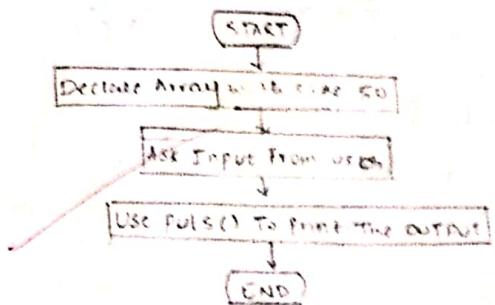
```
#include <string.h>
#include <stdio.h>
#include <conio.h>

void main()
{
    char name[50];
    printf("Enter your name:");
    scanf("%s", name);
    printf("\n your name is %s", name);
    puts(name);
    getch();
}
```

CONCLUSION: Successfully executed the program

print your name (Manisha)  
your name is Manisha

Program



## Ques How to swap

Ans.

(Start)

↓

Declare two variables

↓

Initialise two variables

↓

Use swap function as argument

↓

Print value of variables

↓

(End)

Swap(\*a,\*b);

(Start)

↓

Declare temp variable

↓

Temp = \*a

↓

\*a = \*b

↓

\*b = Temp

↓

(End)

## Practical No: 7.

55

Aim: Program on pointers.

Ques write a C program to Swap two numbers using pointers.

Algorithm:

1. Start the Turbo C application.

2. Declare a function prototype with two integer pointers as arguments before entering main() .

3. Declare two variables and accept their values from the user . Print the respective value using printf()

4. place the address of the variable as arguments for the function

5. Print the respective values of the variable.

6. Use the basic swapping algorithm in the function definition , but instead of using variables use pointers.

Ans # Code:

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int n1,n2;
    clrscr();
    printf("Enter number n1:");
    scanf("%d",&n1);
    printf("\nEnter number n2:");
    scanf("%d",&n2);
    n1=n1+n2;
    n2=n1-n2;
    n1=n1-n2;
    printf("n1=%d",n1);
    printf("n2=%d",n2);
}
```

28

```
scanf("%d %d", &n1, &n2);
printf("\n before swapping: n1=%d, n2=%d", n1, n2)
swap(&n1, &n2);
printf("\n After swapping: n1=%d, n2=%d", n1, n2);
getch();
```

3

```
int swap(int *a, int *b)
{
    int t;
    t = *a;
    *a = *b;
    *b = t;
```

3

CONCLUSION: Successfully executed.

# OUTPUT:

Enter number n1: 21

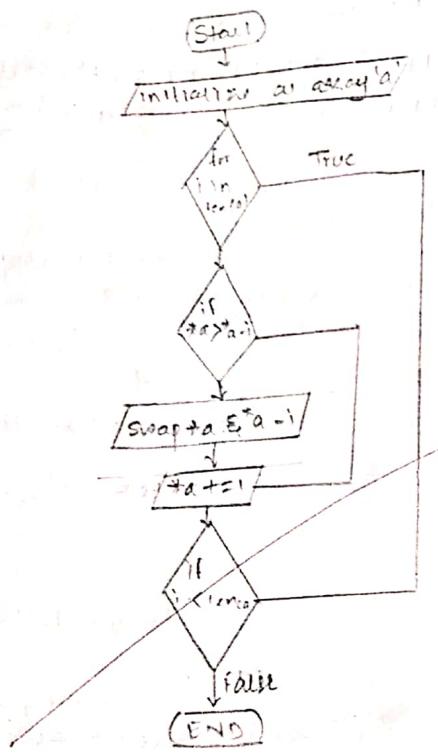
Enter number n2: 23

before swapping: n1=21, n2=23

After swapping: n1=23, n2=21.

56

# flowchart:



(b) Sorting write a c program to sort the array using pointers

Algorithm:

Step1: Initialize an integer array and temp variable

Step2: Run a nested loop of  $i=0$  to  $\text{len}(a)$  and  $j=i+1$  to  $\text{len}(a)-1$ .Step3: If  $*a > *a + 1$ , swap the two value using basic swapping logic.

Step4: print the swapped array.

Step5: terminate the program.

# CODE:

```

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

88

```
int a[10], i, j, temp;  
clrscr();  
for (i = 0, j < 10, i++)  
{  
    for (j = 0, j < 10 - i, j++)  
        if (*a > *a + 1)  
        {  
            temp = *a + 1; *a + 1 = *a;  
            *a = temp;  
        }  
}  
printf("d is the sorted array", a);  
getch();
```

CONCLUSION: Successfully executed.

all output

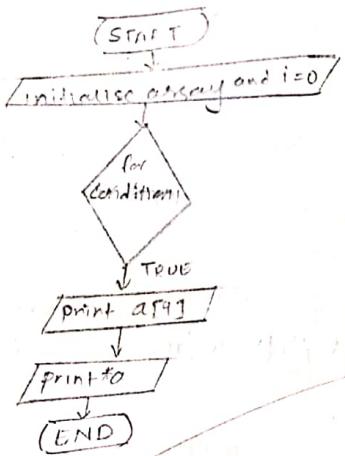
element into the array:

1  
6  
4  
8  
2  
9  
10  
11  
5  
10

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

58

88. # Flowchart



59

e) Aim: write a c program one dimensional array using pointer.

Algorithm:

Step 1: Start the e turbo c application.

Step 2: Initialise the integer array and a variable.

Step 3: Run a for loop with  $i=0$  to the length of array.

Step 4: Print the data of the array and then use pointer to print the array's location.

Step 5: Terminate the program.

#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 the address of a[%d] = %u", i);
        printf("\n the value of a[%d] = %d", i,
```

68

```
int *ptr;  
{  
    int a[5];  
    ptr = &a[0];  
    cout << "The address of a[0] = " << ptr << endl;  
    cout << "The value of a[0] = " << *ptr << endl;  
    cout << "The address of a[1] = " << &a[1] << endl;  
    cout << "The value of a[1] = " << a[1] << endl;  
    cout << "The address of a[2] = " << &a[2] << endl;  
    cout << "The value of a[2] = " << a[2] << endl;  
    cout << "The address of a[3] = " << &a[3] << endl;  
    cout << "The value of a[3] = " << a[3] << endl;  
    cout << "The address of a[4] = " << &a[4] << endl;  
    cout << "The value of a[4] = " << a[4] << endl;
```

CONCLUSION: Successfully Executed.

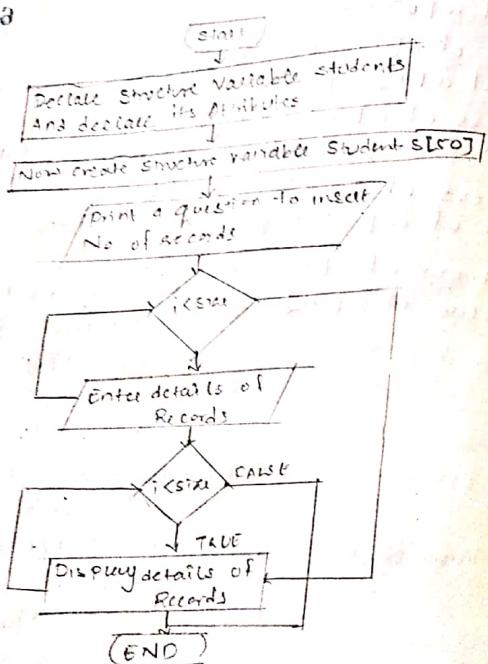
Yours,

OUTPUT

```
The address of a[0] = 65816  
the value of a[0] = 9  
The address of a[1] = 65818  
the value of a[1] = 1  
The address of a[2] = 65820  
the value of a[2] = 4  
The address of a[3] = 65822  
the value of a[3] = 8  
The address of a[4] = 65824  
the value of a[4] = 2
```

Flowchart:

88



PRACTICAL NO:8

61

AIM: Program on Structure and unions

- (a) Create a simple Structure named student that holds the following variable ① id ② CGPA ③ Name

Algorithm:

Step 1: Start the turbo C application

Step 2: Declare a structure variable student and init also declare its attributes i.e id as integers, CGPA as float and name as character datatype

Step 3: Now create structure student S[50] as array and variable size and i as integers

Step 4: Now print a question to ask how many record does user want to be inserted

Step 5: Now use for loop to enter the details of record

Step 6: Again use for loop to display the details of the record.

#include

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

```
struct Student
{
    int id;
    float cgpa;
    char name[30];
};
```

```
struct Student s[50];
int size, i;
printf("Enter how many records you want to\n");
scanf("%d", &size);
for(i=0; i<size; i++)
{
    printf("Enter student ID: ");
    scanf("%d", &s[i].id);
    printf("Enter the CGPA: ");
    scanf("%f", &s[i].cgpa);
    printf("Enter Name: ");
    scanf("%s", &s[i].name);
}
printf("The student record:\n");
printf("\tID\tCGPA\tName\n");
for(i=0; i<size; i++)
{
    printf("\t%d\t%.2f\t%s", s[i].id,
           s[i].cgpa, s[i].name);
}
```

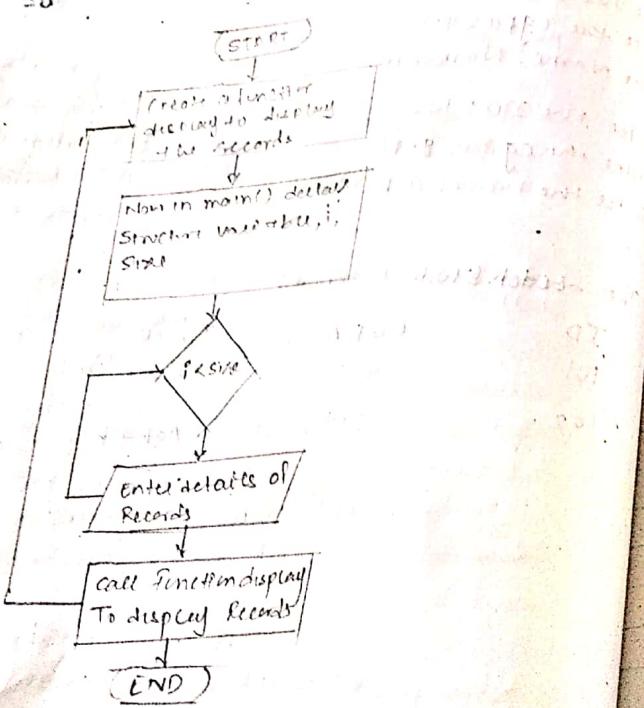
Output

How many record you want to insert : 2  
 Enter the ID: 101  
 Enter the CGPA: 9.3  
 Enter Name: Namrata

Enter the ID: 102  
 Enter the CGPA: 8.9  
 Enter the name: Aakash

The student Record:

ID	CGPA	NAME
101	9.3	Namrata
102	8.9	Aakash



3  
getch();

conclusion: program executed successfully

(b) write a program which will demonstrate the use of function and structure

Algorithm:

Step 1: Start turbo C application

Step 2: Declare a function display with argument as struct as struct student &s[50]

Step 3: Now in main() declare i, size as integer, structure variable student s[50]

Step 4: Now use for loop to insert the detail of records

Step 5: Now call the function display to display the record

Step 6: End the program.

\* core

\* include <stdio.h>

\* include <conio.h>

void display (struct student s[10])

void main()

{

    struct student

    { int roll;

        char name[50];

    };

    struct student s[10];

    int i, size;

    printf("How many records you want to input")

    for (i=0; i<size; i++)

    {

        printf("The Roll No:");

        scanf("%d", &s[i].roll);

        printf("Name:");

        scanf("%s", &s[i].name);

    }

    display(s);

    getch();

}

void display (struct student s[100])

    int i, size;

    printf("Display record:");

    printf("1. R. N. 2. Name");

18

OUTPUT:  
How many record do you want to insert : 2

The Roll no : 101

Name : Neha

The Rollno : 102

Name: Mayer

Display Records!

RN	Name
101	Neha
102	Mayer

65

```
for(i=0; i<size; i++)
```

```
    printf("\n%1d %s\n", s1[i].roll, s1[i].name);
```

3

conclusion: successfully executed the program.

- o) create union to store data of a student in the form of rollno, stud-name, div, percentage, contact-no. Insert data and print the same.

Algorithm:

Step 1: Start turbo c application

Step 2: Declare union variable student and init also declare rollno and contact-no as integer and array, percentage as float and div and stud-name as character datatype

Step 3: Now create union student s[50] and declare size as integer -

Step 4: Now use for loop to insert the details of records.

Step 5: Now use for loop again to insert details  
seconds

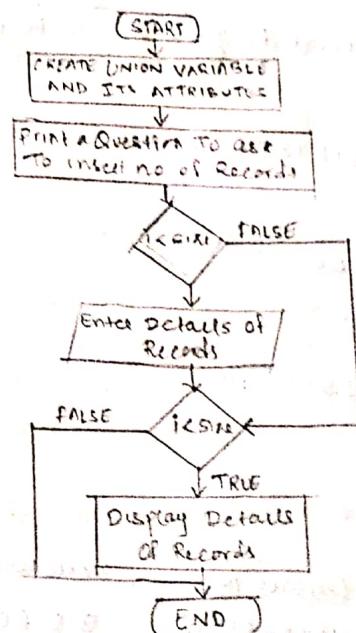
Step 6: End the program

~~Code:~~

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    Union student
    {
        int roll_no, contact_no [10];
        float percent;
        char division, name [30];
    };
    Union student s [50], int i, size;
    printf ("How many seconds do you want to
    insert?:");
    scanf ("%d", &size);
    for (i = 0; i < size; i++)
    {
        printf ("\nEnter Rollno:");
        scanf ("%d", &s[i].roll_no);
        printf ("\nEnter contact no:");
        scanf ("%d", &s[i].contact);
        printf ("\nEnter Name:");
        scanf ("%s", &s[i].name);
        printf ("\nEnter division:");
        scanf ("%c", &s[i].division);
    }
}
```

flowchart

Flowchart



43  
OUTPUT:  
How many record do you want to insert: 2

Rollno: 101  
Contactno: 9898989898  
Name: Nakul  
Division: A  
percentage: 88.60

Roll No : 102  
Contact No : 8888899999  
Name : Haresh  
Division : B  
Percentage : 90% 18

RN	Name	Div	Contact No	Percentage
101	Nakul	A	9898989898	88.60
102	Harsik	B	8888899999	90.18

67

```

printf("In percentage : ");
scanf("%f", &s[i].percentage);
}

printf("In IERNt name \tdiv \tcontent no
       \tpercentage in");
for (i = 0; i < size; i++)
{
    printf("\n\t%.d \t%.s \t%.c \t%.d \t%.f
          s[i].collect.name, s[i].name, s[i].div,
          s[i].content_no, s[i].percentage);
}

getch();
}

```

Conclusion: program executed successfully.

58

### PRACTICAL NO. 9

AIM: programs on string manipulation

A) Write a program to copy one string into another string

Algorithm:

Step 1: Declare a 2 character array and i as integer.

Step 2: Ask user to enter a string.

Step 3: Use while loop to check null character in characters array to copy the two arrays.

Step 4: Now check null character for 2 string Variable to terminate the loop.

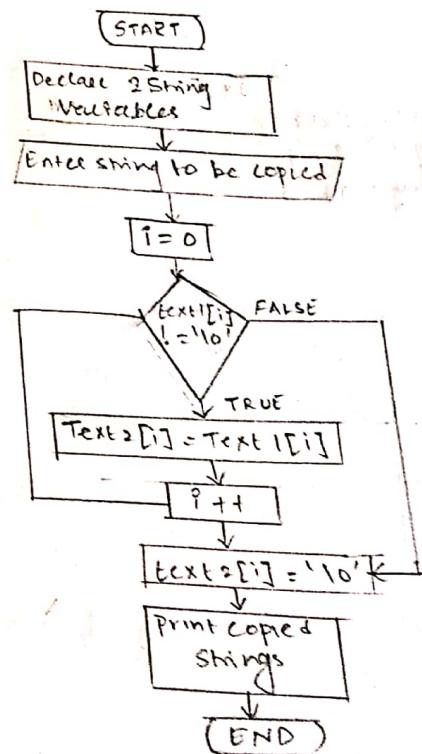
Step 5: Print copied strings

Step 6: End the program.

#CODE:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char text1[50];
    char text2[50];
    int i;
    i = 0;
```

PROBLEMS



OUTPUT:

Enter a string: abcd  
 First string: abcd  
 Second string: abcd

```
printf("Enter a string:"),
scanf("%s", &text1);
while(&text1[i] != '\0')
{
    text2[i] = text1[i];
    i++;
}
if(text2[i] == '\0');
printf("First string= %s\n", text1);
printf("Second string= %s", text2);
getch();
```

CONCLUSION. Program executed successfully.

15

(c) write a program which displays the length of a string without using string function.

Algorithm:

Step 1: Start turbo c application

Step 2: Declare a character array, \*t & and len=0 as integers

Step 3: Now enter a string

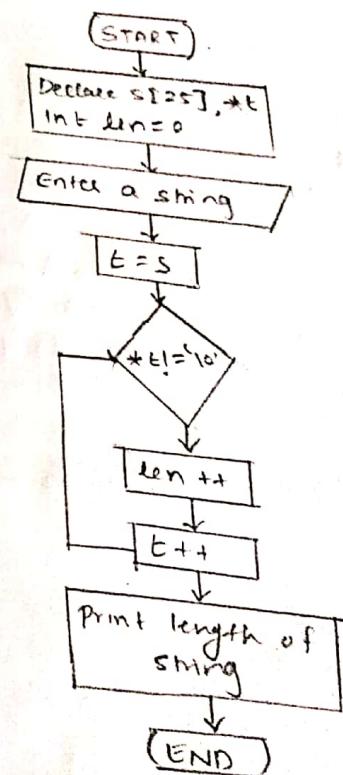
Step 4: Use t=s to copy base address of a string

Step 5: use while loop to count length of string

Step 6: print length of string

Step 7: End the program.

FLOWCHART



87  
# OUTPUT:  
Enter a string: Hello  
Length of string is 5

73

#CODE:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char s[25], *t;
    int len = 0;
    printf("\nEnter a string:");
    scanf("%s", &s);
    t = s;
    while (*t != '\0')
    {
        len++;
        t++;
    }
    printf("Length of a string is %d", len);
    getch();
}
```

Conclusion: Program executed successfully!

## PRACTICAL NO. 10

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

Algorithm:

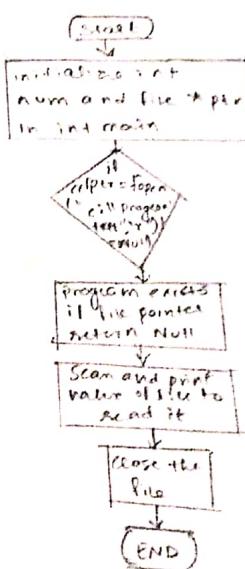
- f.open() → opens a existing file/create a new file
- f.read() → Reads a record from a file
- f.close() → close a file

Code for reading from a text file/opening/closing.  
Text file is C.txt and its contents are: 87 88 89 90

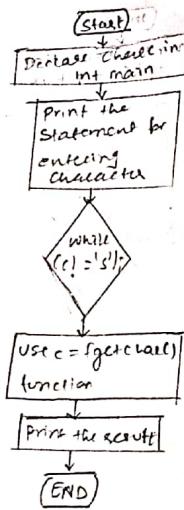
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int num;
    FILE *Fptr;
    if((Fptr = fopen ("C:\1\program.txt", "r")) == NULL)
    {
        printf("Error: opening file");
        exit(1);
    }
    fscanf(Fptr, "%d", &num);
    printf(" Values are = %d ", num);
    fclose(Fptr);
    return 0;
}
```

read from text file or close it

flowchart:



OUTPUT - Values are - 87 88 89 90



# OUTPUT:

Enter some character Enter \$ to exit

A

Entered Character is : A

B

Entered Character is : B

Entered character is : \$

(b) Aim: Write a program for fgetc(), fgetchar(), fputchar()

Algorithm

fgetchar is a file handling function.

It is used to read a single character from key word input

Code

```

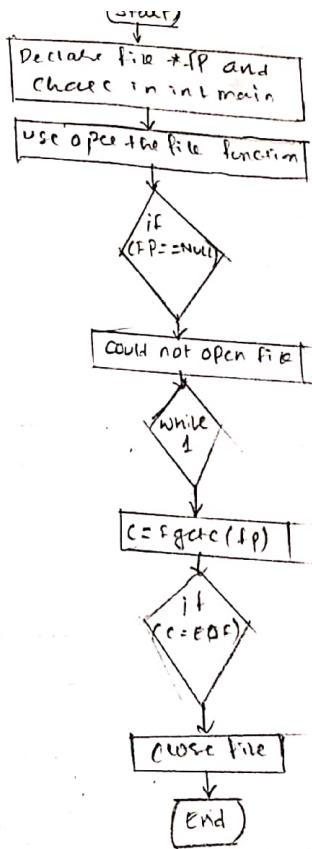
#include <stdio.h>
#include <ctype.h>
int main()
{
    char c;
    printf("Enter some Character Enter $ to exit");
    while(c != '$');
    {
        c = fgetchar();
        printf("\n Entered character is : ");
        putchar(c);
        printf("\n");
    }
    return 0;
}
  
```

- 85
- c) `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:

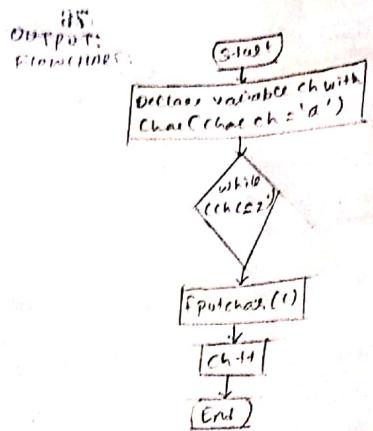
```
#include <stdio.h>
int main()
{
    file *fp;
    char c;
    printf("Opening file test.c in read mode");
    fp = fopen("test.c", "r");
    if (fp == NULL)
    {
        printf("could not open file test.c");
        return 1;
    }
    printf("Reading the file test.c");
    while (1)
    {
    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

Procedure:



Output:

Opening the file test.c in read mode  
 Reading the file test.c  
 Hi, how are you?  
 Closing the file test.c



OUTPUT: abcdefghijklmnopqrstuvwxyz.

close(2);  
return 0;

3

- (b) `putchar()` - a file handling function is used to write a character on a standard output / screen.  
 If `putchar()` function is equivalent to `putchar()` function where `char` is a character variable.

#Code:

```

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

```

Conclusion: Thus all the programs are successfully executed.