C PROGRAMMING: PRACTICAL 1

(C To Display Name, Roll No, Percentage)

AIM: Write a C program to understand basic data type and input/output

REQUIREMENTS: Turbo C

ALGORITHM: STEPS:

This C program creates a file and store information
We frequently use files for storing information which can be processed by our program
In order to store information permanently and retrieve it we need to use files and this program demonstrate file creation and writing data in that
The source code is used to create a file and store information
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

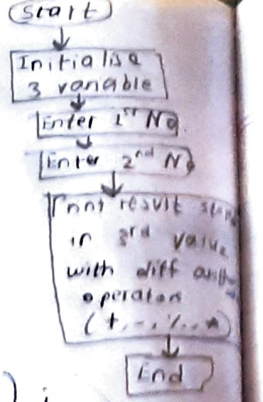
Verified

OS Program:

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

Output:

```
Enter your roll number: 1889
Enter your percentage: 88
Enter your name: Ayaan
Enter your mobile number: 8452xxxxxx

Your roll number is: 1889
Your Percentage is: 88
Your name is: Ayaan
Your Mobile number is: 8452xxxxxx
```



TS C PROGRAMING : PRACTICAL 2

Aim: Write a C program on operators and expressions

Requirements: Turbo C

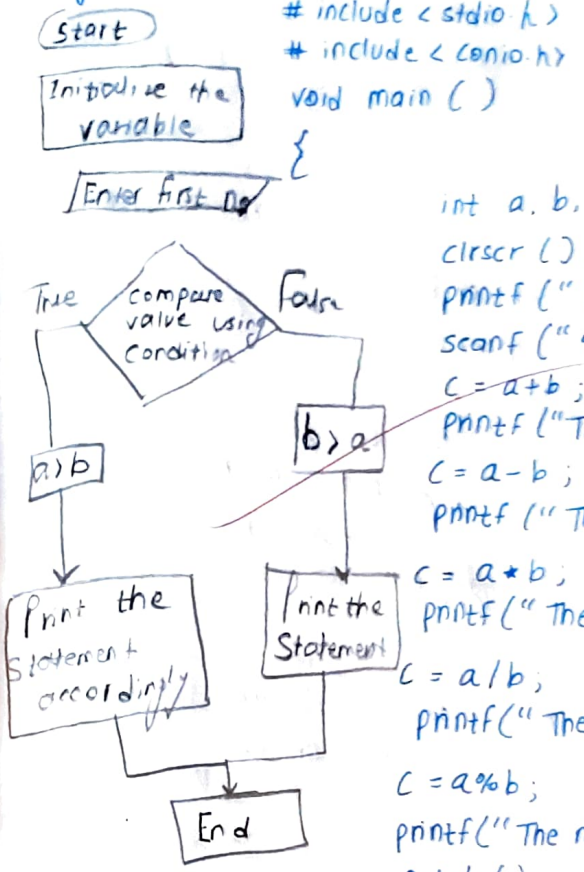
→ Operators are the foundation of any programming language. Thus the functionality of C language is incomplete without the use of operators.

→ Operators allows us to perform different kinds of operations on operands.

* Anthmetic Operators

1. Addition : The '+' operator adds two operands (a+b)
2. Substraction : The '-' operator substracts 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 modulo of the nos is %d \n ", c) ;
getch ( ) ;
}
```

Output :

```
Print two numbers : 20 10
The sum of the nos is 30
The diff of the nos is 10
The mult of the nos is 200
The div of the nos is 2
The modulo of the nos is 0
```

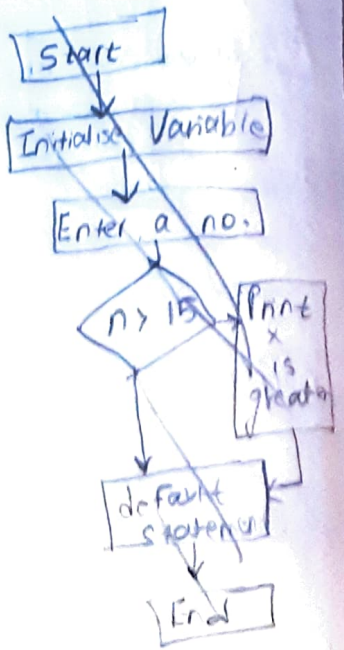
Q8

Program Code:

```
#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



Algorithm

Step 1: Declare Variables a, b & 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 print function to display output.

Conclusion: These programs help us in having better understanding about operators and expressions.

PRACTICAL 3 - C PROGRAMING.

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 condition that 20 is less than 15 and if condition is false skip the if statement and print I am not in if.

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

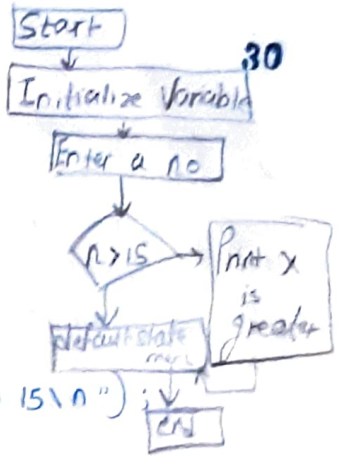
Step 1: Declare a variable as integer and assign its value ie 20.

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

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

a] Code: If statement

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

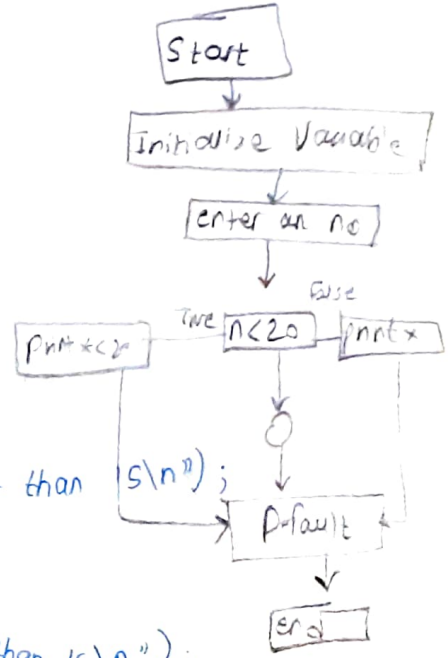


output:

I am not in if.

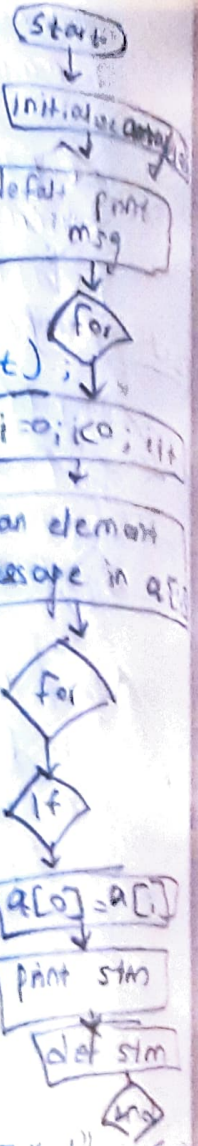
b] Code: If else statement

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



output: 20 is greater than 15.

```
#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[i]);
  getch();
}
```



output:

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

The largest number is 100.

Aim: C program to find largest array number using array

Algorithm:

Step 1: Start turbo c 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 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.

Aim: WAP to print the even and odd numbers in an array.

Algorithm:

Step 1:- Create an array taking size from user and define an array using loop

Step 2:- Display the size of the array

Step 3:- Display the element of array entered by user.

Step 4:- Take the initiator in a for loop using which all the elements in the array

Step 5:- Display even no from array from for loop
If (array[i] %.2 == 0)

Step 6:- Display the even no. from the given array

Step 6:- Display the odd number
if (array[i] %.2 != 0)
Display odd no from array.

Step 7 - Close or terminate turbo C.

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

```
    printf("%d", array[i]);
    }
    getch();
}
```

output :

Enter the number of elements : 5

1, 4, 6, 8, 9

Even no : 4

6

8

Odd no : 1

9

iii.

Algorith :

Step 1 : Start turbo C application

Step 2 : Initialize the int variable a[100], num i, and float variable sum = 0 and average.

Step 3 : Start the user for the length (<100) and store the value using scanf.

Step 4 : Put avg = sum / num

Step 5 : Print the value of sum & avg.

Step 6 : Terminate the program

Source code ↘

Output : → Next page

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; i++)
      for ( i=0 ; i< num ; i++)
      {
          printf ("\n")
          scanf ("%d", &a[i]);
          sum + k a [i];
      }
      avg = sum / num ;
      printf ("\n the sum of no. is %f" sum, avg);
      getch() ;
}
```

Conclusion: The program is successfully executed

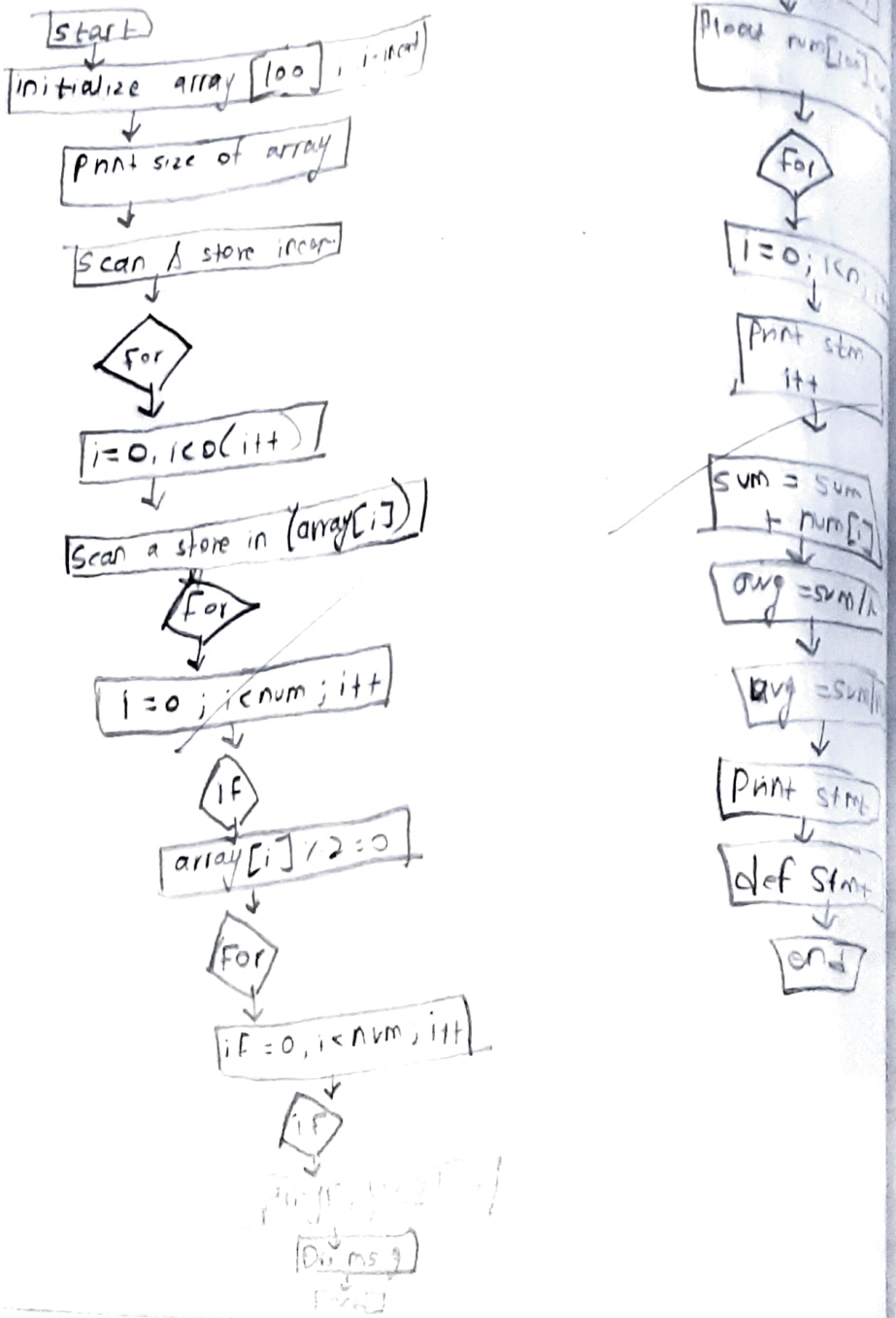
Learned

output:

enter the no of element 10

2
3
6
5
10
1
2
1
7
11

The sum of the no. is 48.0000
and the avg is 4.80000



PRACTICAL NO 6.

Aim: Factorial of no. using recursion and use of get().

Algorithm :-

Step 1 :- Start the turbo c application.

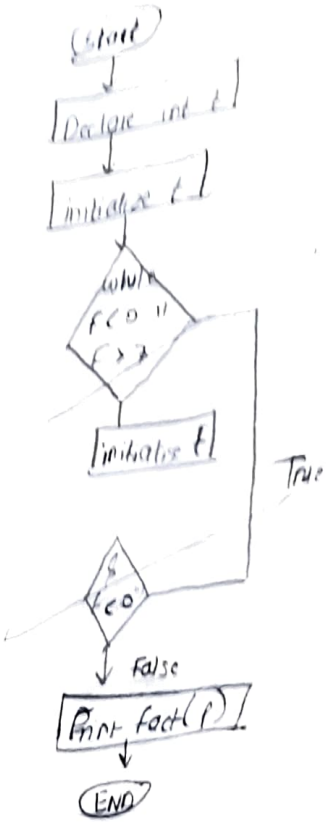
Step 2 :- Declare a function prototype entering the main() function

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

Step 4 :- Use the while function to ensure value entered in the range 0 to 7.

Step 5 :- Print the value entered by the function in step 02.

Step 6 :- Terminate the program



output

Enter a number of factorial (0,7) = 4. 24

source code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int f
    clrscr();
    printf("enter a no to('0.to7'));
    scanf("%d", &f);
    while(f<0||f>7)
    {
      printf("enter a no in range 0 to7")
       scanf("%d" &f)
    }
    printf("%d", fact(f));
}
int fact(int n)
        if (n>=1)
        {
            return n* fact(n-1);
        }
        else
        {
            return 1;
        }
```

ii. Algorith

Step 1 Start the turbo C application

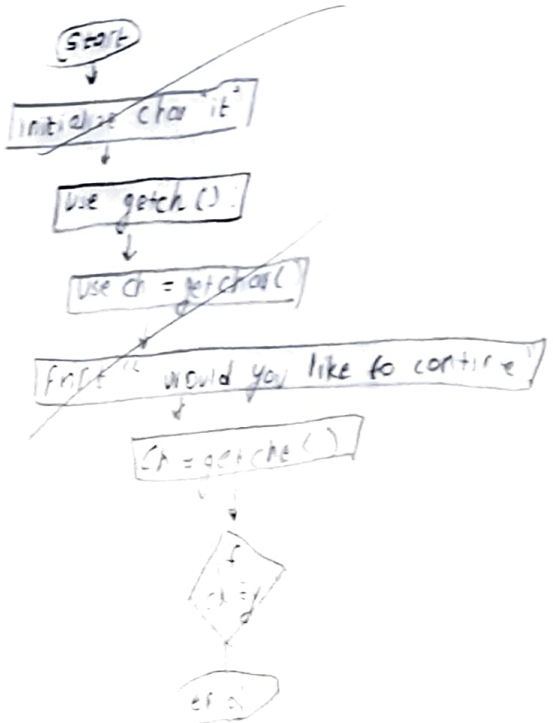
Step 2 Initialize a character variable 'ch'

Step 3. Use the getch() method to accept the character.

Step 4 Use the getchar() method to store a 'y' or 'n' in 'ch'.

Step 5: while ch == 'y', keep accepting values fo

Step 6: Terminate the program



Output :-

press any key to continue

Enter any character = A

would you like to continue (y/n) ? = y

would you like to continue (y/n) ? = n.

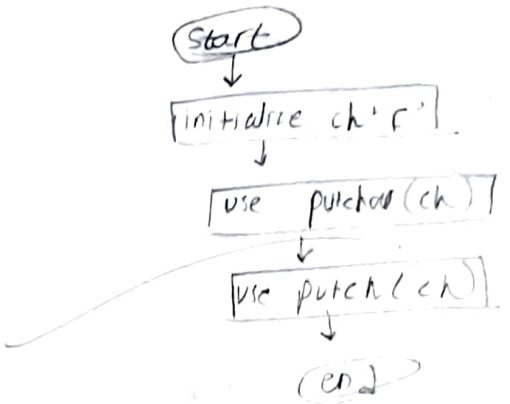
Source code :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
  char ch: 'n'
   clrscr();
   printf(" press any key to continue :);
   getch();
   printf("\n enter any character ');
   ch = getchar();
   printf("would you like to continue (y/n)");
   ch = getche();
   while = (ch='y')
   {
    printf("would you like to continue ?");
   }
getch();
}
```

Ahmadi

Algorith :-

1. Start the turbo C application.
2. Initialize a character ch to 'n'
3. Use the putch() and putchar() function with ch as the argument.
4. Terminate the program



Output

A

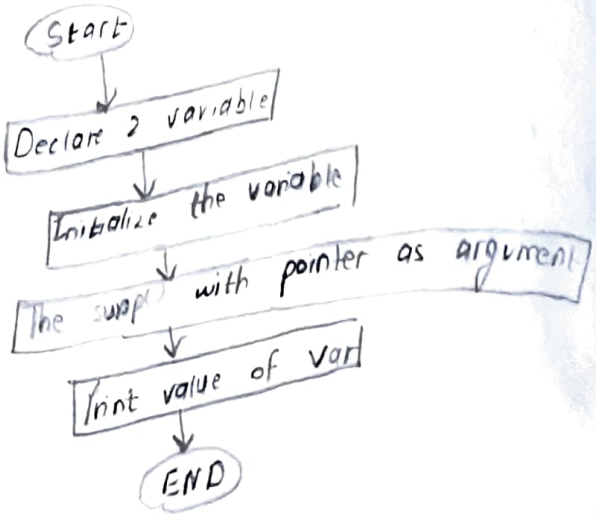
A

SA

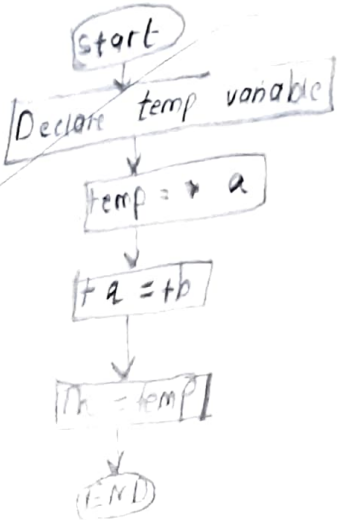
Source code:

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

Conclusion:- The cause of factorial use of getch() and put function have been done successfully



swap(*a,*b):



Aim :- Write a program to find the Swapping of 2 Nos. (Pointers)

Algorith :-

step 1 - Start the turbo :- C 7 application.

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

Step 3 - Declare 2 variable and accept thus value from the user, print the reputive value using printf()

Step 4 - Pair the address of the variable as argument for the function

Step 5 - Print the repective value of the variable.

step 6 - Use the basic swapping algonthm in the Paction definition but instead of normal valuable use.

Source code:

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

    getch();
}
void swap(int *m, int *n)
{
    temp = *m;
    *m = *n;
    *n = temp;
}
```

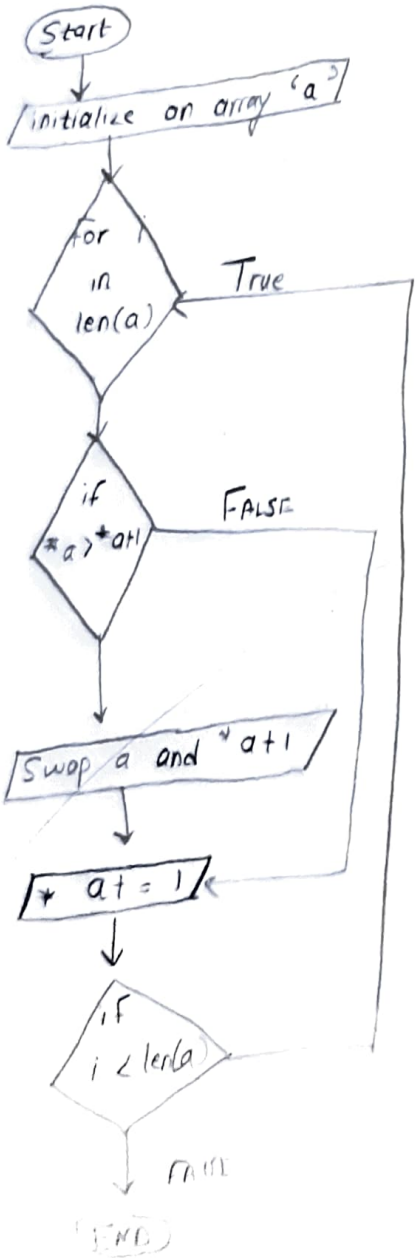
output:

Enter the 2 no. to be swapped 12

24

The no before swapping are 12 and 24

The no after swapping are 24 and 12.



Sorting of arrays using pointer

Algorith

Step 1: Initialize an integer array, 'i' and temp variable

Step 2: Run a nested loop of i=0 to len(a) and of i=0 to len(a) - i

Step 3: if *a > *a+1, swap the two values using basic swapping logic.

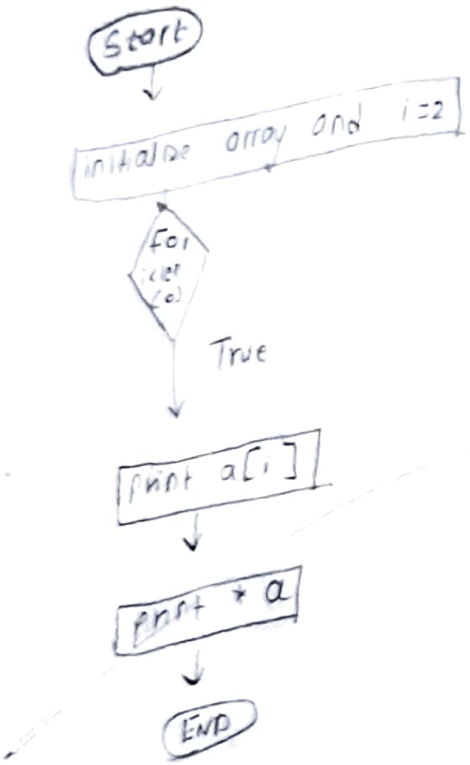
Step 4: Print the swapped array

Step 5: Terminate the program

elements into the array

8
6
4
3
1

1, 2, 3, 4, 5, 6, 8, 9 is the sorted array



Write a program to find one dimensional array using pointer.

Algorithm :

Step 1 : Start the turbo C 7 application

Step 2 : Initialize an integer array and a variable.

Step 3 : Run a for loop with a = 0 to length of array

Step 4 : Print the data of the array and then use pointer to print the memory location

Step 5 : Terminate the program.

* Code:

```
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] = %u", i, ptr);
    printf(" The value of a[%d] = %d", i, *ptr);
    ptr++;
    i++;
}
getch();
}
```

Conclusion: The program to find one-dimensional array using pointer is done successfully.

Khwaja

Output:

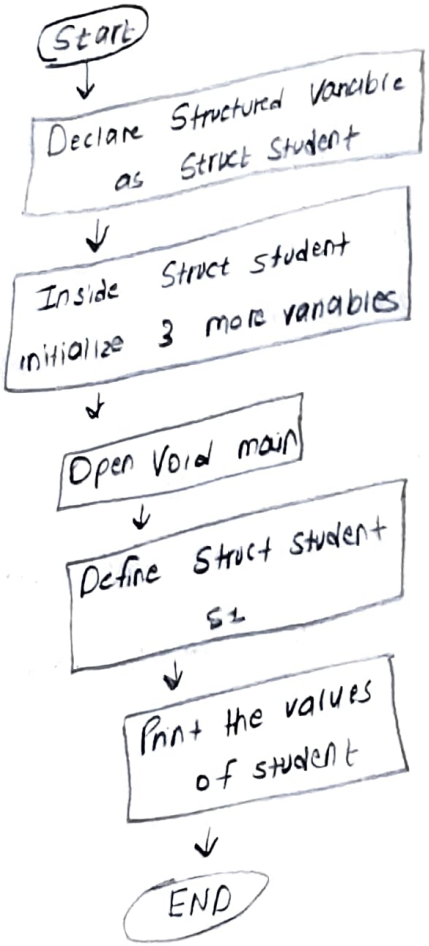
The address of a[0] = 65516
The value of a[0] = 7

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



A).

Aim: Create a simple structure named as Student that holds Following vanable : id, CGPA, Name.

Algorith:

Step 1: Start turbo C 7 application.

Step 2: Declare the structured vanable as 'Struct -Student'.

Step 3: Initialize the struct student with 3 more vanables inside it as 'int id', 'float CGPA', 'char name[10]'.

Step 4: Now, inside void main() define struct student S1;

Step 5: Print the details of the student such as id, CGPA, Name.

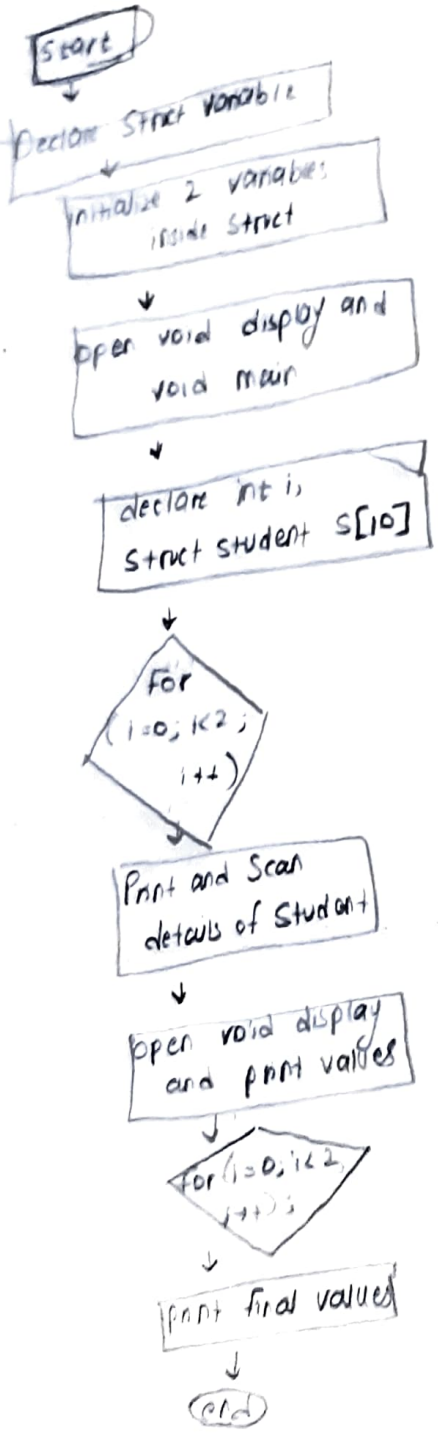
Step 6: Terminate the program.

Source code:

```
struct student
{
  int id;
  float CGPA;
  char name[10];

};

void main()

{
    struct student S1;
    printf("Enter id, CGPA and name of student");
    scanf("%d %f %s", &S1.id, &S1.CGPA, S1.name);
    printf("\n id = %d", S1.id);
    printf("\n CGPA = %f", S1.CGPA);
    printf("\n name = %s", S1.name);

}
```

Output:

```
Enter id, CGPA and name of student
1 8.333 Ayaan
 id = 1
CGPA = 8.333
Name = Ayaan
```



8.

Aim: WAP which will demonstrate use of structure and function

Algorithm:

Step 1: Start turbo C 7 application

Step 2: Declare the structured variable as struct student

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

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

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

Step 6: Print the details of students

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

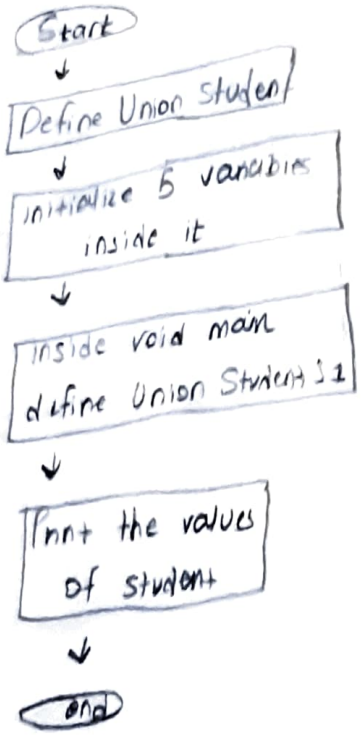
Step 8: Terminate the program.

Code:

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

output:

```
Enter details of 2 student
Enter roll and name 22 Om
Enter roll and name 33 Prakash
********
Roll = 22    Name = Om
Roll = 33    Name = Prakash.
```



C.

Aim: Create union to store data of a student in the form of roll no, stud name, div, Percentage, contact no, insert the data and print it in suitable manner in a command.

Algorithm:

1. start turbo C 7 application
2. Define the union as student (Union Student)
3. Initialize the union student with 5 variables namely roll no, stud name, div, Percentage, Contact no
4. Now, inside void main function define Union Student ;1.
5. Print the values of the student
6. Terminate the program

#Code

```
#include <stdio.h>
#include <conio.h>
Union Student
{
    int roll;
    char stud name[10];
    char div[5];
    float percentage;
    long contact no;
};
void main()
{
  Union Student S1;
  clrscr();
  S1.stud name, "Ayaan";
  printf("Name = %s", S1.stud name);
  S1.roll = 55;
  printf("Roll no = %d", S1.roll);
S1.div, "A";
  printf("Div = %s", S1.div);
S1.percentage = 85
printf("Percentage = %f", S1.Percentage);
S1.contact no = 8452041111
printf("Contact = %ld", S1.contact no);
getch()
}
```

Output

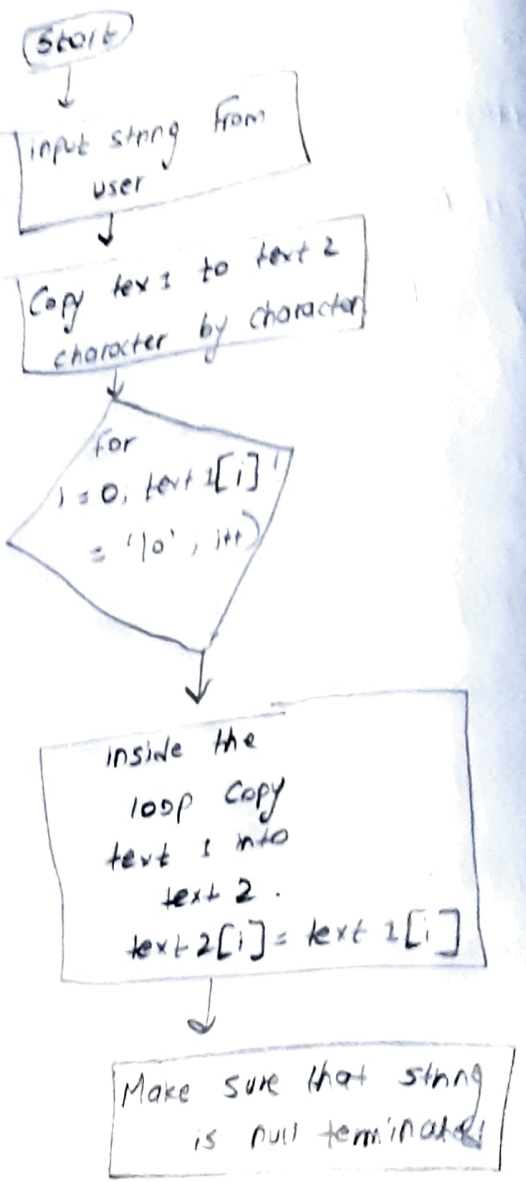
name Ayaan

Roll no 55.

Percentage 85.

Contact no. 8452041111

Div A



Aim WAP to copy one string into another string

Algorithm

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

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

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

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

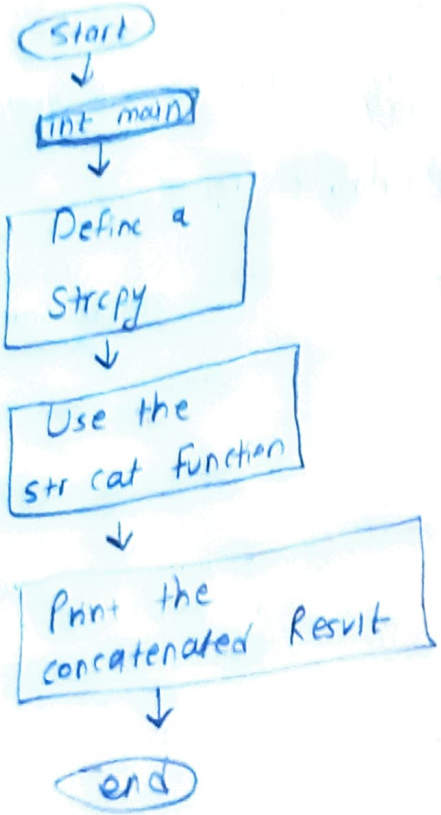
Step 5: Finally after loop make sure the copied string ends with NULL character i.e. text2[i] = '\0';

Code.

```
#include <stdio.h>
#define MAX_SIZE 100

int main()
{
 char text1[MAX_SIZE];
 char text2[MAX_SIZE];
 int i;
 printf("Enter any string: ");
 gets(text1);
 for(i=0; text1[i]!='\0'; i++)
 {
     text2[i] = text1[i];
 }
 text2[i] = '\0';
 printf("First string = %s\n", text1);
 printf("First string = %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  There are 7 days in a week
Total character copied  26.
```



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

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

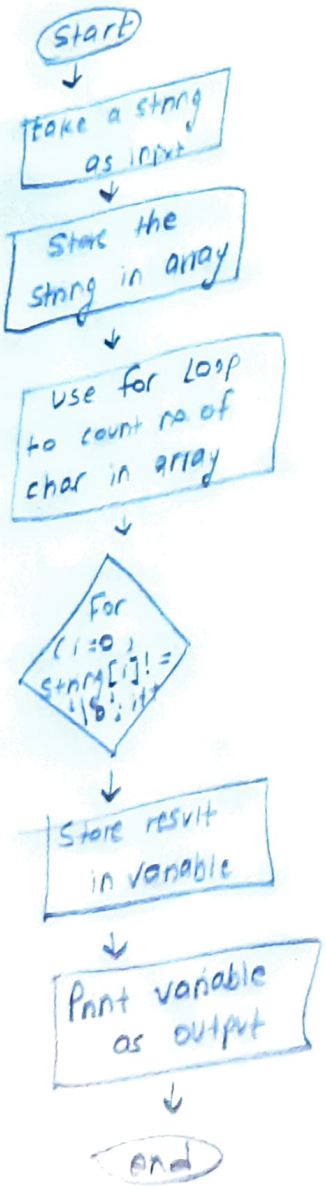
Code:

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

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

output Rahul is over 18 years old

80



C. 1

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

Algorithm:

1. Take a string as input and store it in the array
2. Using for loop count the number of characters in the array and store the results in the variable.
3. Print the variable as output.

12

Code:

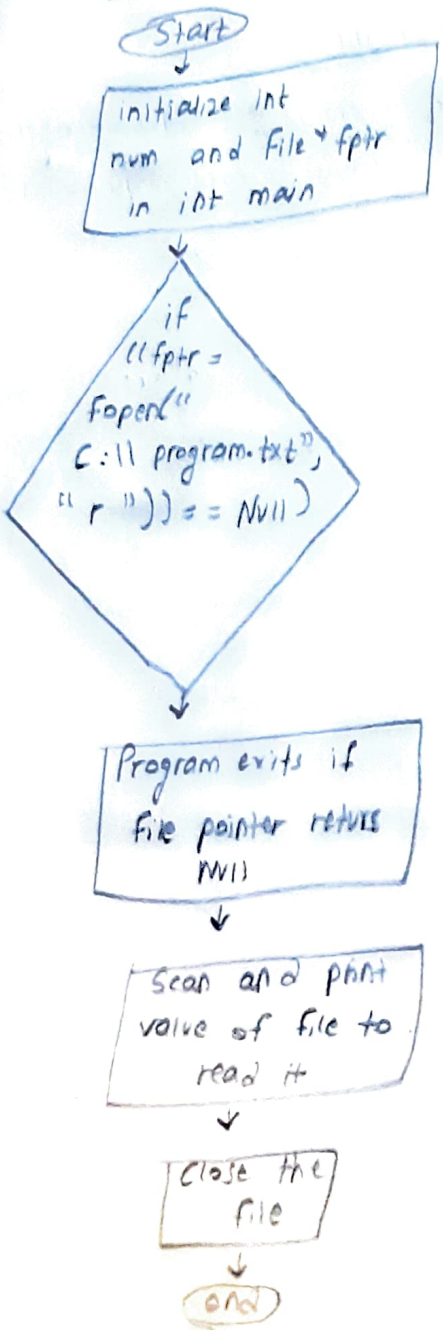
```
#include <stdio.h>
void main()
{
    char string [50];
    int i, length = 0;

    printf(" Enter a string \n");
    gets(string);

    For(i=0; string[i] != '\0'; i++)
    {
        length ++;
    }

    printf("The len of str is the no. of charaters in
    printf("so the len of: %s = %d \n", string, leng
```

output

```
Enter a String:
  It is a cold night
```

The length of str is the no. of characters in it
so the length of: It is a cold night = 18.

Read from text file and close it.



A.

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

Fopen() → Opens a existing file or create a new file for use.

Fread() → Reads a Record from a file.

fclose() → Closes a file.

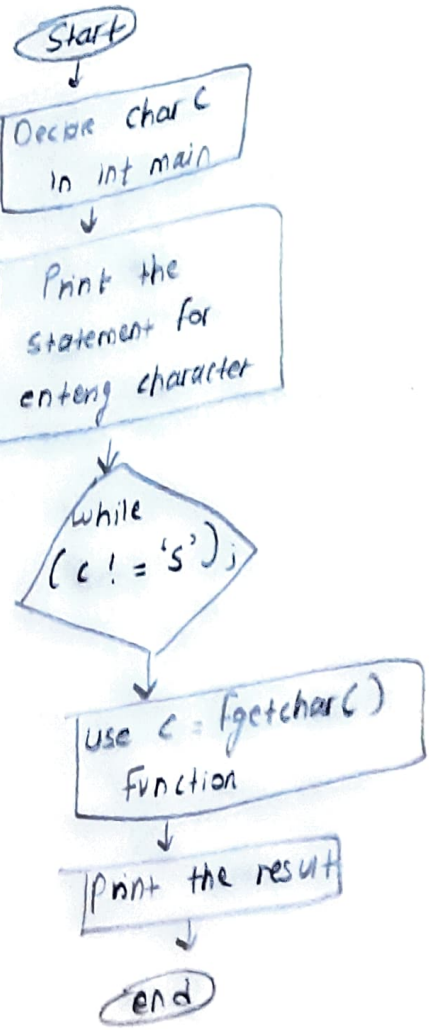
code for reading from a text file / opening / & closing.
Text file is (.txt and its contents are : 87
88
89
90

```
# include <stdio.h>
# include <stdlib.h>
int main ()
{
   int num;
   FILE * fptr;

   if ((fptr = fopen ("C:\\ program.txt", "r")) == NULL)
    {
       printf ("Error ! opening file");

       exit(1);
    }
   fscanf ( fptr, "%d", &num );

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

   return 0;
   }
```

output :

values are = 87
88
89
90

ia



B. Aim : WAP for fgetc(), fgetchar(), fputchar() function.

Algorithm & Description.

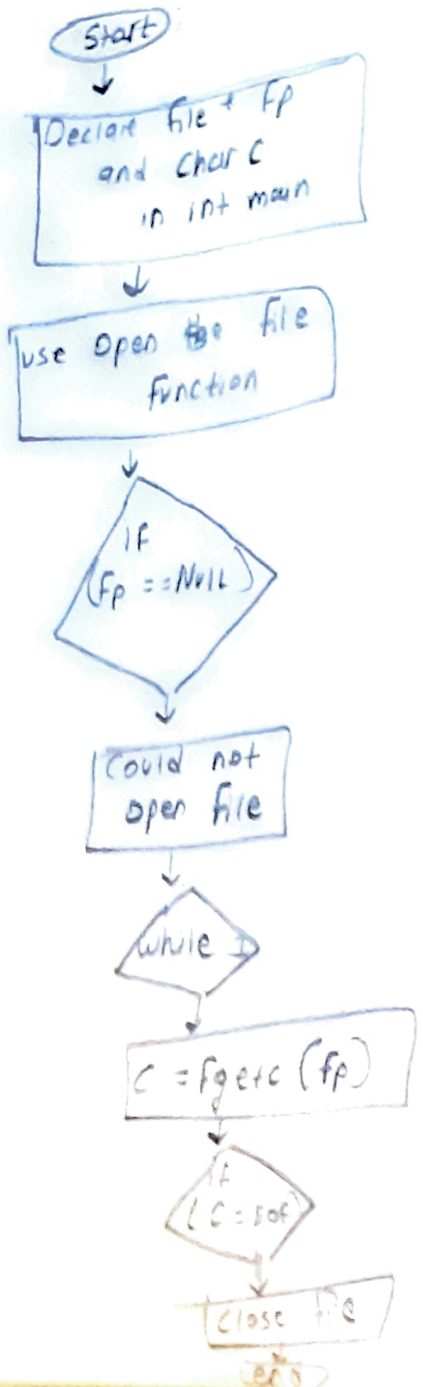
- Fgetchar is a file handling function
- It is used to read a single character from keyboard input.

Code:

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

    }
    return 0;

}
```

Output:

```
Enter some character. Enter $ to exit
A
Entered character is: A
B
Entered character is: B
$
Entered character is $
```



fgetc() → Used to read a character from a file.
Reads single character at a time.
In a program we use fgetc() function

```
fgetc(fp);
```

where

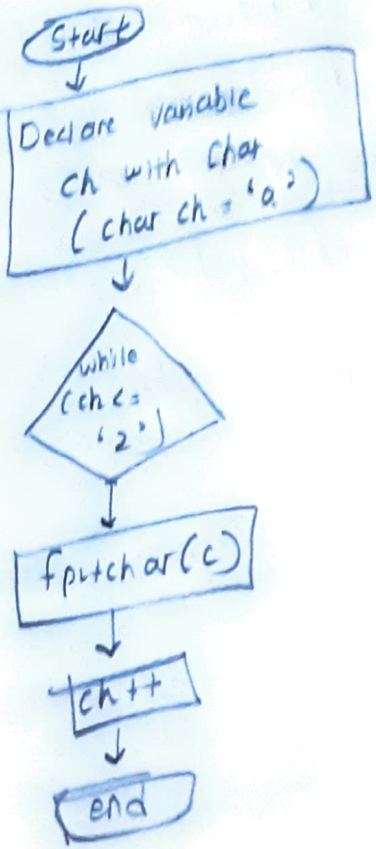
fp = file pointer.

Code :

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

output:

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



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

Code :

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

Output : abcdefghijklmnopqrstuvwxyz.