

Practical 1:

Aim: Write a C program to understand basic datatypes & I/O.

Theory:

Write a program to display student's name, roll no, mobile no & percentage.

Algorithm:

Step 1: Declare a variable name, roll no as integer, also declare name, mobile no as character & percentage as float

Step 2: Use printf function to print question 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.

28

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int roll_no;
    char name[20], mobile_no[10];
    float percentage;
    clrscr();
    printf("enter student's name:\n");
    scanf("%s", &name);
    printf("enter student's roll no:\n");
    scanf("%d", &roll_no);
    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 roll no: %d\n", roll_no);
    printf("student's mobile no: %s\n", mobile_no);
    printf("student's percentage: %f\n", percentage);
    getch();
}
```

88.

Output:

Enter student's name:

Lalit

Enter student's rollno:

1880

Enter student's percentage:

78.80

Enter student's mobile no:

8888888888

Student's Name: Lalit

Student's rollno: 1880

Student's mobileno: 8888888888

Student's percentage: 78.80

29

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

Nimedi

P.S Practical 2:

Q1] Write a C program on operators and expression.

~~Theory~~

A.] Write a program to create a dynamic calculator.

~~Algorithm.~~

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

Step 2: Now use scanf function to receive input from user.

Step 3: Now to add two numbers given by user, use the expression

num1 + num2;

Step 4: Now to subtract two numbers

given by user expression

num1 - num2.

Step 5: Again we expression num1 * num2 of user

with user to multiply the two

Step 6: Use expression num1 / num2 if user

wishes to divide the two inputs.

START

30

Initiate the variable

Enter first number

Enter second number

Arithmetic Expression

Print results

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int num1,num2;
    float add,sub,mult,div;
    clrscr();
    printf("Enter first number:\n");
    scanf("%d", &num1);
    printf("Enter second number:\n");
    scanf("%d", &num2);
    odd = num1 + num2;
    sub = num1 - num2;
    mult = num1 * num2;
    div = num1 / num2;
    printf("Addition of %d and %d is %f\n", num1, num2, add);
    printf("Subtraction of %d and %d is %f\n", num1, num2, sub);
    printf("Multiplication of %d and %d is %f\n", num1, num2, mult);
    printf("Division of %d and %d is %f\n", num1, num2, div);
    getch();
}
```

Output:

Enter first number=3

Enter second number=3

Addition of 3 and 3 is 6.0000

Subtraction of 3 and 3 is 0

Multiplication of 3 and 3 is 9

Division of 3 and 3 is 1.0000

Step 7: Now use print function to display output.

B) Write a program in C to implement ternary operator.

Algorithm:

Step 1: Declare variables a, b & c as integers.

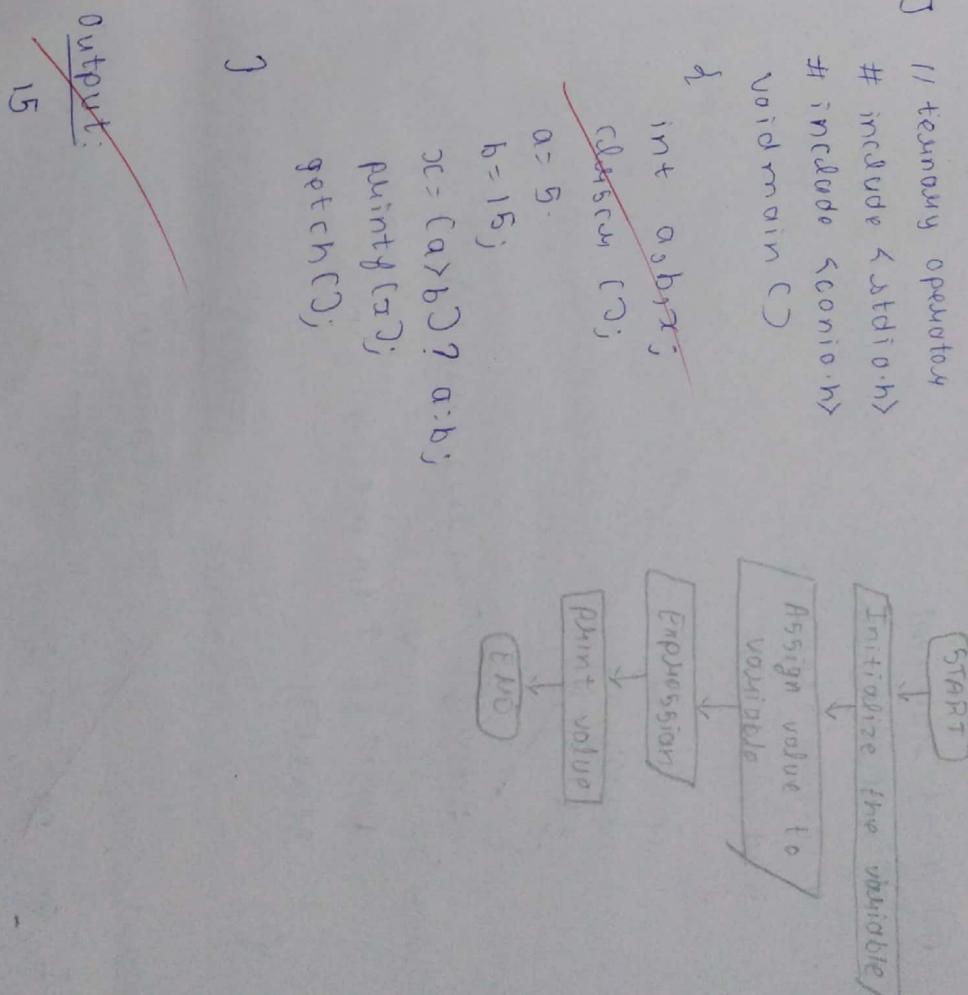
Step 2: Store the value of a as 5 & store the value of b as 15.

Step 3: Use to compare between who is greater use ternary operator to find

Step 4: Use printf function to display output.

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

Nimisha



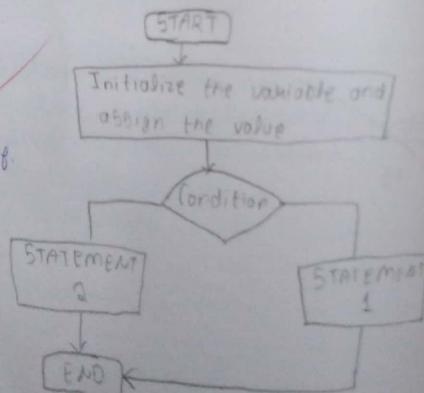
```

A) // 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.

Practical 3:

Ques: Write a program in C 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 10 to 20

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

Step 3: If the condition is true, printf that 20 is less than 15 & if condition false skip if statement & print I am not in if.

88.

B] write a program in C to ascertain if else statement.

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

Step 2: You to compare the given value if it's greater or not and use if else conditional statement

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

34.

B) If else statement

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

```
int i=20;
```

```
clrscr();
```

```
if (i<15)
```

```
printf("20 is smaller than 15\n");
```

```
}
```

```
else
```

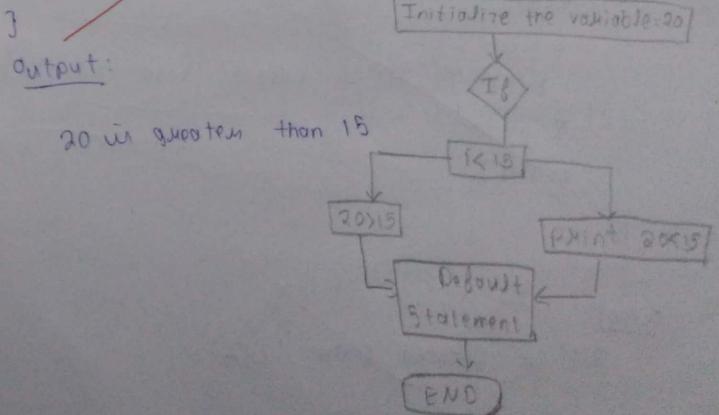
```
{ printf("20 is greater than 15\n");
```

```
}
```

```
getch();
```

Output:

20 is greater than 15



C-J // nested if

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

```
void main()
```

```
{ int i=20;
clrsgr();
if (i<15),

```

```
if (i<12)
{

```

```
if

```

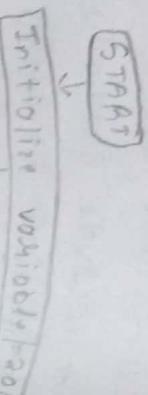
```
print ("20 is less than 15 & 12\n");
}
else:

```

```
}

```

```
{ print ("20 is greater than 15 & 12\n");
}
getch();
```



C-J Write a program in C to explain nested if statement.

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

Step 2: Now we need if logic to compare if given no is greater or not

Step 3: If first condition is true then go to the second condition if second condition is also true then print that 20 is greater than

& 15 & 12. If one of the conditions are not true then skip the part & print 20 is greater than 15 & 12.

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

Answe

Output:

20 is greater than 15 & 12.

A] Qn: To display the prime numbers using for loops.

Algorithm

Step 1: Initialize the variables out of which two are loop variable and one are count variable.

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

Step 3: Nest another loop within the loop in step 2 that goes to 2 to the first variable x_2 .

Step 4: Use the if condition statement to check whether (first loop variable % 2nd loop variable) is 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.

Step 6: Terminate the program.

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int n, i, a
    clrscr();
    printf("The prime numbers are:");
    for (i=2; 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

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

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

```
clrscr();
```

```
printf("Enter number of elements\n");
```

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

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

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

```
{ n3=n1+n2;
```

```
printf("%d",n3);
```

```
n1=n2;
```

```
n2=n3;
```

```
}
```

```
getch();
```

Output:

Enter number of elements 10

0 11235813213455

Ans. ENO.

Conclusion: Thus, we have successfully executed fibonacci

series in Turbo C

b] Sum: Write a C program to find sum of Fibonacci series

Algorithm:

Step 1 Start the Turbo C

Step 2 Declare the variables n1, n2, n3, i, number

Step 3 Initialize the variable n1=0, n2=1 & number=0

Step 4 Enter the no. of terms of fibonacci series to be printed

Step 5 Print first two terms of series as

n1=0 & n2=1

Step 6 Use the for loop as per following step:

n3=n1+n2

n1=n2

n2=n3

Step 7 Increase the value of i elements each time by 1.

Step 8 Print the value of number.

c.) Ques: 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

Step 2: Declae the variable n_{rows}; i_j; j_{number} = 1

Step 3: Display the no. of rows

Step 4: Enter the no. loop: i_j < n_{rows}; i_j++

Step 5: Create nested for loop: i_j = 1; j_{number} = i_j; j_{number}++

Step 6: Display the no. as per user enter the required
from i=1.

Step 7: Increment numbers from 1.

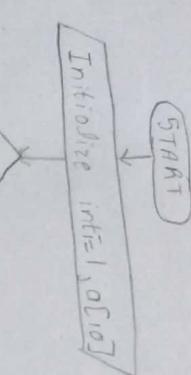
Step 8: Display the space

Conclusion: Thus, we have successfully executed given
expression on Turbo C using nested for loop

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



Ques: Write a C program to find the largest array number using array.

Algorithm:

Step 1: Start twelve C application.

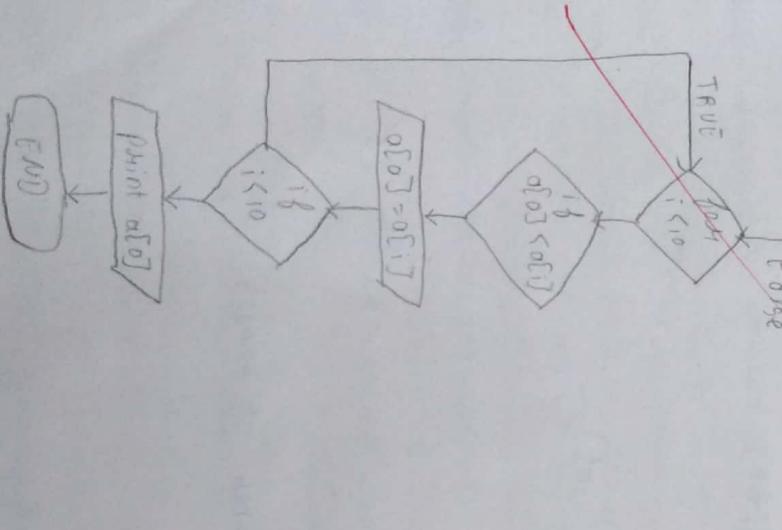
Step 2: Declare the variables ; and integer array $a[10]$

Step 3: Enter the for loop at $i=0$, $i \leq 10$ and use the value of $a[i]$ ≤ 10 . End the for loop.

Step 4: Enter the for loop at $i=0$, $i \leq 10$ 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 $i \leq 10$, exit the loop.

Step 6: Terminate the program.



Program

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

```
void main()
```

```
{ int a[10], i;
```

```
clrscr();
```

Prints the element of the list \n.

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

```
{
```

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

```
}
```

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

```
{
```

```
if (a[i] < a[i+1])
```

```
{
```

```
a[0] = a[i];
```

```
}
```

```
}
```

```
printf("The longest number is %d", a[0]);
```

```
 getch();
```

```
}
```

Output:
Enter the elements:

12
13
2
12
13
55
3
22
100

The largest number is 100.

Conclusion: Through this program we have successfully executed largest always leaving away number

number

Q3 Write a C program to find even and odd numbers.

using away

Algorithm:
Step 1: Create an array, take its size from user & define
+ print using loop.

Step 2: Display the size of array entered by user.

4.3. Define the element of society entered by man

~~Take the initiative in & for your writing which all~~

Steps: Display the even no. of array from first to last

then display it is a user re

~~Step 6: Display the odd no. of array from user
for (array[i] % 2 == 0)
then display it is a odd no.~~

Step 7: Terminate the process.

780

Program

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

```
void main()
```

```
{ int arry[100],j,num;
```

```
clrscr();
```

```
printf("Enter the size of the array\n");
```

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

```
printf("Enter the element of the array\n");
```

```
for(j=0;j<num;j++)
```

```
scanf("%d",&arry[j]);
```

```
printf("Even no. of the array\n");
```

```
for(j=0;j<num;j+=2)
```

```
if(arry[j]%2==0)
```

```
printf("%d\n",arry[j]);
```

```
if(arry[j]%2!=0)
```

```
printf("%d\n",arry[j]);
```

```
printf("In odd no. in the array are:\n");
```

```
for(j=0;j<num;j+=2)
```

```
if(arry[j]>arry[j+2])
```

```
printf("%d\n",arry[j]);
```

```
printf("%d\n",arry[j+2]);
```

Output

Enter the size of the array 4

Enter the element of array

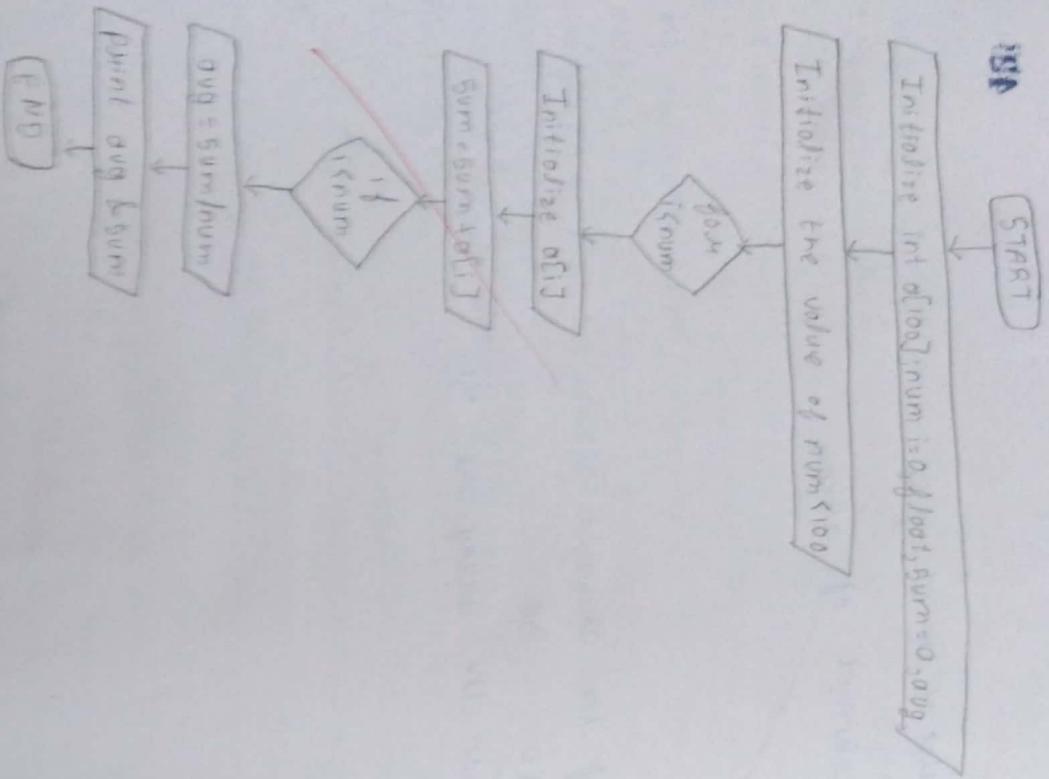
10
20
30

35

Even no. of the array

10
20
30

Odd no. in the array are: 35



C) sim C program to find average and sum using array

Algorithm:

Step 1: Start Turbo C application

Step 2: Declare the float int variable n, initialize sum=0.0f, sum=0.0,

Step 3: Avg = sum / n at i=0; i<n; i++ ; give print message and increment by 1.

Step 4: Below sum variable and Avg at Avg adding num[i],

Step 5: Average is sum divided by n.

Step 6: Give exit statement for average and sum.

Step 7: terminate the program.

Enter no. of elements : 10

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

```
int n;
```

```
float num[100], sum=0.0, avg;
```

```
clrscr();
```

```
printf("Enter the no. of elements");
```

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

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

```
printf("Enter the no. %d ", i+1);
scanf("%f", &num[i]);

```

```
sum = sum + num[i];
}
```

```
avg = sum / n;
```

```
printf("Average = %.2f", avg);

```

```
printf("Sum = %.2f", sum);
getch();
```

The sum of the numbers is 48.000000
and the average is 4.80000

Conclusion: Through the program we have successfully created the program for largest number, addition, numbers and sum/average of an array.

Unacademy

Ques: To find factorial of a number using recursion

Algorithm:

Step 1: Start Turbo C application.

Step 2: Declare the int variable factorial, n.

Step 3: Use if conditional statement and return factorial, and use else statement for returning.

Step 4: Declare int variable n, 0

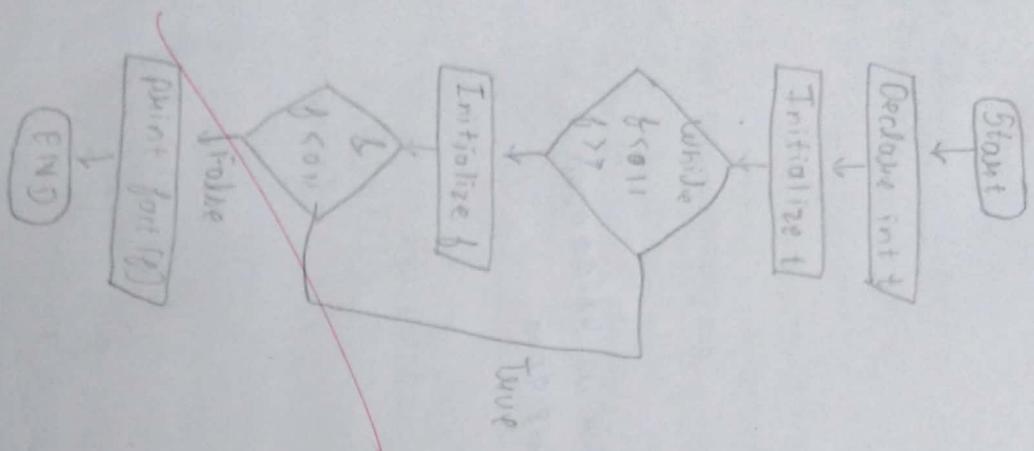
Step 5: Use print statement for taking input from user.

Step 6: Factorial of n is a

Step 7: Use default statement

Step 8: Display the output

Step 9: Terminate the program.

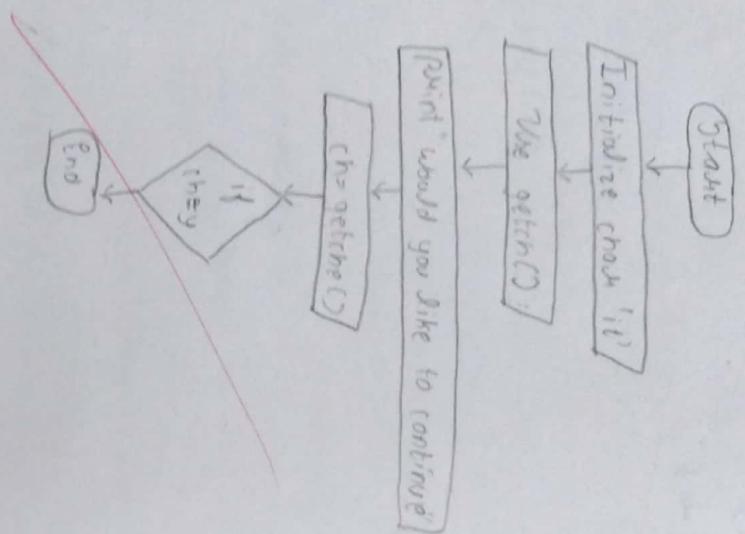


Program:

```
#include <stdio.h>
#include <conio.h>
void factorial(int);
void main()
{
    int a;
    clrscr();
    printf("Enter number:");
    scanf("%d",&a);
    factorial(a);
    getch();
}
void factorial(int n)
{
    int i,fact=1;
    for(i=1;i<=n;i++)
    {
        fact=fact*i;
    }
    printf("Factorial of %d is %d",n,fact);
}
```

Output:

Enter number:3
Factorial of 3 is 6



b) Write a program which draws the use of `get()` function.

Algorithm:

Step 1: Start the Turbo C application.

Step 2: Initialize a character variable 'ch'.

Step 3: Use the `getch()` method to read the character.

Note: Use the `getch()` method to store 'a' or 'n' in ch

Step 4: While `ch == 'y'`, keep accepting values for ch

Step 5: Terminate the program.

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char ch;
    clrscr();
    printf("Press any key to continue");
    getch();
    printf("Would you like to continue (Y/N)?");
    while (ch != 'y')
    {
        printf("Would you like to continue (Y/N)?");
        getch();
    }
}
```

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

Start

Initialize (r)

Use getch()

Use putch(r)

END

(J)

Algorithm:

Step 1: Start the Turbo C application.

Step 2: Initialize a character to 'r'

Step 3: Use the putch() & getch() function with ch as the argument.

Step 4: Terminate the program.

Program:

#include<conio.h>

#include<stdio.h>

void main()

{

char ch='A'

clrscr()

putch(ch)

printfl("n")

putchar(ch);

getch();

}

~~Ques~~ Conclusion: The use of factorial, use of getch() & put

() have been done successfully.

Step: Write a program to find the sum of Two Numbers.

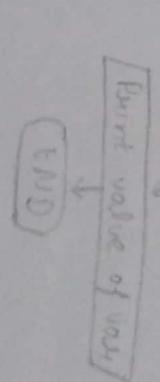
Algorithm:

Step 1: Start the ~~Two Number Summation~~ Application.

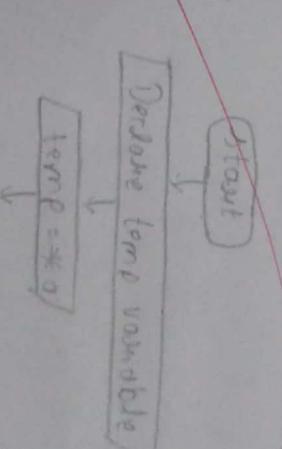
Step 2: Declare a function prototype with two integer pointer as argument before entering main()?

Step 3: Declare two variable and ~~accept~~ the value from the user, Print the value using printf()

`scanf("%d %d",`



Step 4: Pass the address of the variables as argument from the function.



Step 5: Print the received value of the variable.

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

`END`

Output:

Enter the 2 nos. to be swapped: 12, 24
 The no. before swapping are 12 and 24
 The no. after swapping are 24 and 12.

Program:

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

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

Conclusion: The program swapping of Two Numbers is successfully done.

Sorting of array using partition.

Algorithm:

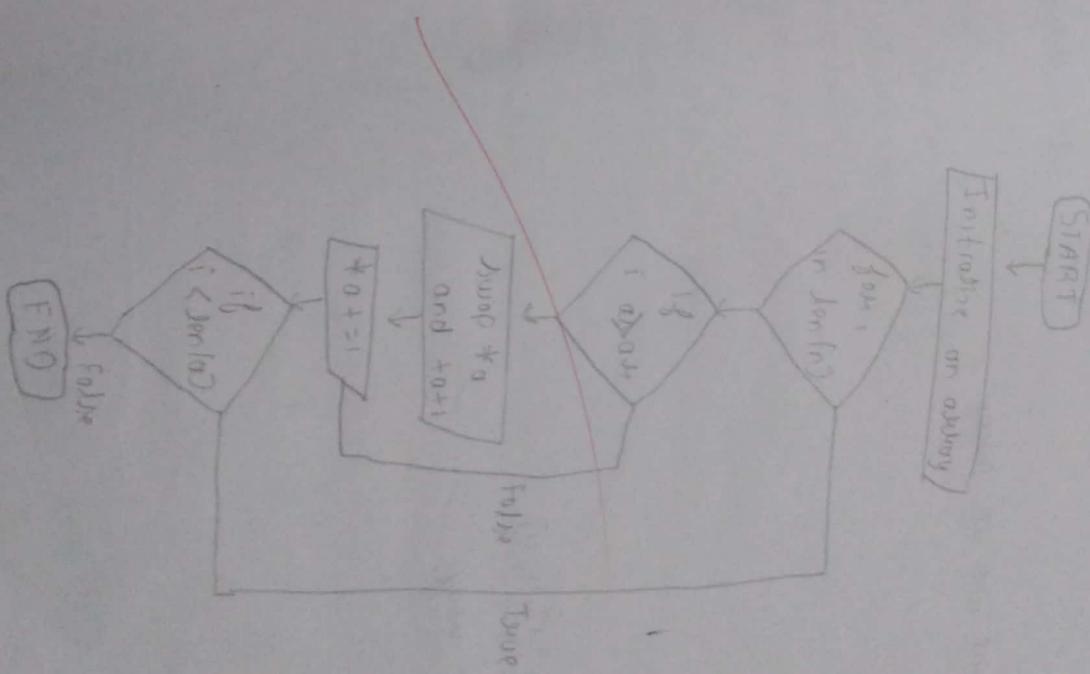
Step1: Initialize an integer array and temp variable.

Step2: Run a nested loop of for $i=0$ to $\text{len}(a)-1$ and

Step3: $y = a[i+1]$, swap the two consecutive values using basic swapping logic.

Step4: Print the swapped array.

Step5: Terminate the program.



Output:

Insert element into the array

1

6

7

8

2

9

3

4

5

10

11

12

13

14

15

16

17

{ 1,2,5,6,7,8,9,10,11,12 }

is the sorted array.

Source code:

```

Code
void sort(int n, int *p)
{
#include <stdio.h>
#include <conio.h>
Void main()
{
    int a[10], i, 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\n", *a);
    getch();
}

```

Conclusion: The program sorting of array using pointer is done successfully.

iii] Write a program to find one-dimensional array using pointers

→ Algorithm

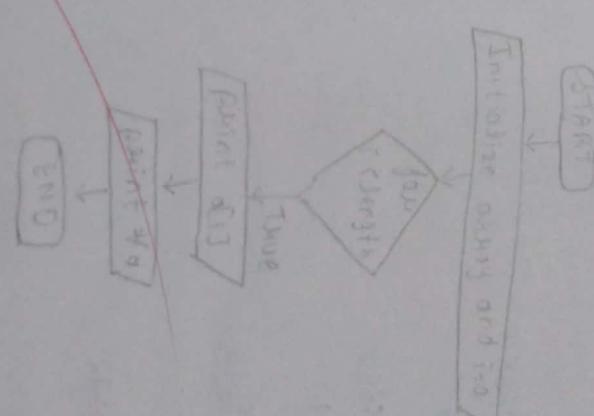
Step 1: Start the Turbo C application.

Step 2: Initialise an integer array and all variables

Step 3: Run a for loop with $i=0$ 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



Output:

The address of a[0]=65516

The value of a[0]=7

The address of a[1]=65518

The value of a[1]=9

The address of a[2]=65520

The value of a[2]=4

The address of a[3]=65522

The value of a[3]=8

The address of a[4]=65524

The value of a[4]=2

Source code:

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

```
void main()
```

```
{ int a[5] = {7,9,4,8,2};
```

```
int *ptr;
```

```
int i=0;
```

```
ptr=&a[0];
```

```
clrscr();
```

```
while(*ptr != '\0')
```

```
{ printf("\n\n The address of a[%d] = %u , %p\n"),
```

```
pointf["\n The value of a[%d] = %d , %p\n"],
```

```
ptr+i,
```

```
i+1,
```

```
}
```

```
getch();
```

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

Answe

Q8]

Ques: Create a simple structure that holds following
variable : id, CGPA, Name

Algorithm:

Step: Start the tuple C application

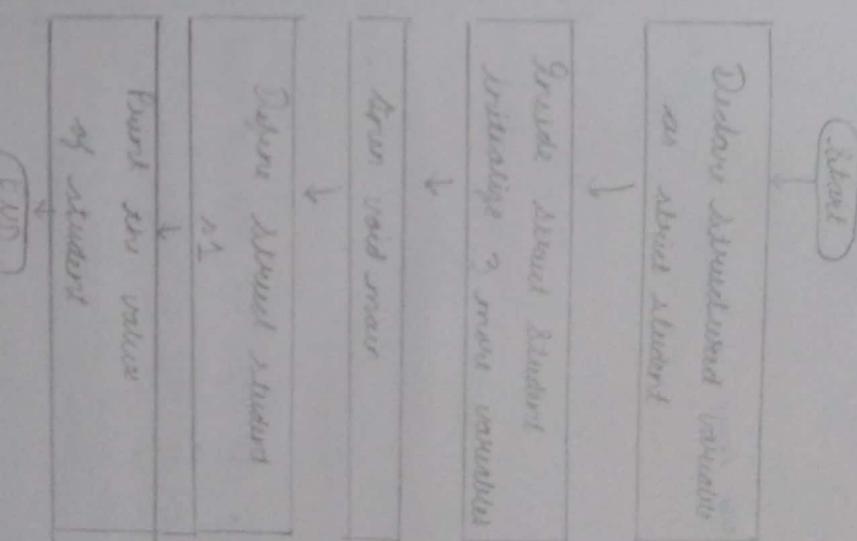
Step2: Declare the structured variable as 'struct-student'

Step3: Initialize the struct student with 3 more
variables inside it as 'int-id', 'float CGPA', 'char
Name[10]'.

Step4: Now create void main() define struct student

Step5: Print the details of the student like as
id, CGPA, Name

Step7: Terminate the program.



Output:

Enter id, CGPA and name of student

id = 1

CGPA = 8.373

Name = Lalit

Source code:

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

void main()
{
    struct Student st;
    printf("Enter id, CGPA and name of student");
    scanf("%d %f %s", &st.id, &st.CGPA, st.name);
    printf("\nId = %d", st.id);
    printf("\n CGPA = %.2f", st.CGPA);
    printf("\n name = %s", st.name);
}
```

6] Arm W.A.C.P which will concentrate all of attention

End function

Annotation

Sign: Stand the Task C Application

Step 3 Initialize the Street Stewart with two α 's.

\rightarrow Fig 2: Reduce the structured variables as structureless

Next visitors will main display and visit main lecture and is about student [unclear]

Jesus: Use the few brief few writing details of student notes in students and not much more.

Ans. Print the details of the Student

Step 3: Run and display again and start the value using condition & switch

With the exception of

Output:

59

Enter details of a student

Enter roll and name 8080 Lalit

Enter roll and name 90 Shubham

* * * * *

Roll = 80 Name = Lalit

Roll = 90 Name = Shubham

Code:

```
#include <stdio.h>
struct student
{
    int roll;
    char name[10];
};

void display();
void main()
{
    int i;
    struct student s[10];
    display();
    printf("\nEnter details of a student");
    for(i=0; i<2; i++)
    {
        printf("\nEnter roll and name");
        scanf(" %d %s", &s[i].roll, s[i].name);
    }
    display();
}

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

Practical

Ques. Write a program to copy one string into another string.

Solution:

Steps: Input string from "User" & store it in 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++)`

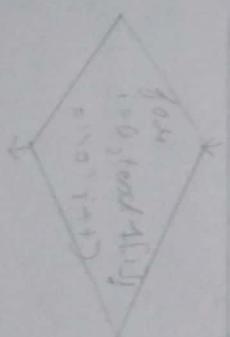
Step 4: Inside the loop for each character in text 1 copy to text 2. Say `text2[i] = text1[i]`

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

Flow

Input string
from user

Copy text 1 to text 2
char by char



Inside the loop
copy text 1
into text 2.

End text 2

Make user wait
to exit

Make user wait
to exit

Make user wait
to exit

Output:

Enter any string : There are 7 days in a week
 First String : There are 7 days in a week
 Second String : There are 7 days in a week

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++)
    {
        int a[i]=text1[i];
    }
    text2[i]='\0';
    printf("First string = %s\n",text1);
    printf("Second string = %s\n",text2);
    printf("%s\n", "Total character copied = %d\n");
    return 0;
}
```

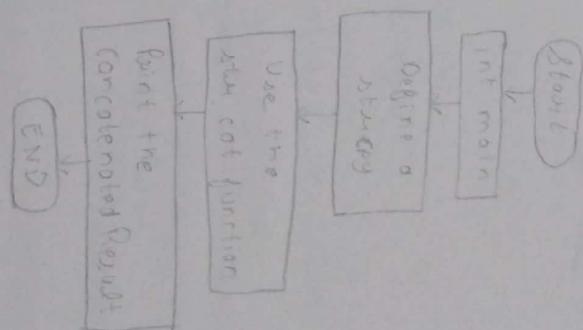
Q] W.A.P which will demonstrate the use of string library function.

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

1) dest

2) str

The strcat() returns a pointer.



Output: Rabat is over 19 years old.

Code:

```
#include <stdio.h>
#include <string.h>
{
    int main(int argc, const char * argv[])
{
    char example [100];
    strcpy(example, "Rabat");
    printf("%s\n", example);
    return 0;
}
```

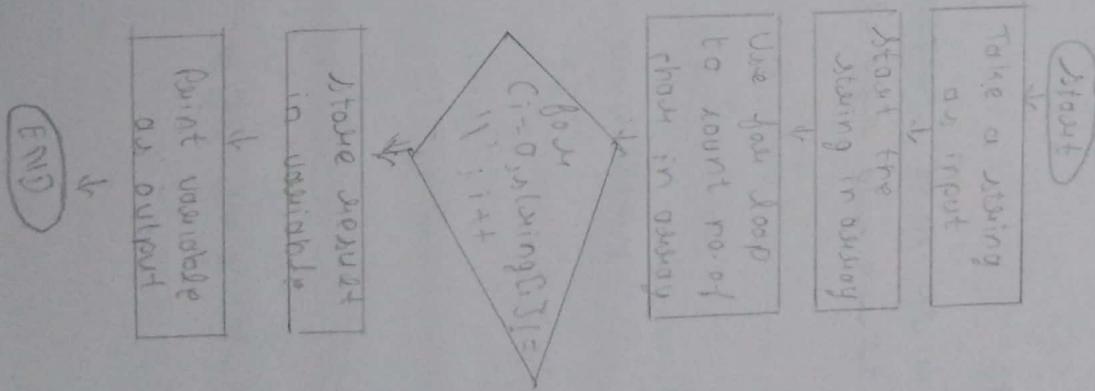
C.J
WAP which displays the length of a string without using string function.

Algorithm:

Step: Take a string as input and store it in the array.

Step: Using for loop count the number of characters in array & store the result in the variable.

Step: Print the variable as output



Ansatz:

It is a cold night

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

Solu

#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 characters is %d,\n",
printf ("%s is the len of : %d", string, length);

A) Aim: Program for file open, read & close.

Input() → opens a existing file or choose a new file for use.

fread() - Read a record from a file.

fclose → close a file

Blant

```
Initialize int
numby file & fp in
int main
```

if fopen
("file", "r") = NULL

Program exits
file pointer becomes
null

Scan & print value of
file to read it

↓

close the file

END

test file in C:\ext & its contents are

Abdullah
 Value one: 97
 68
 56
 40

Code

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

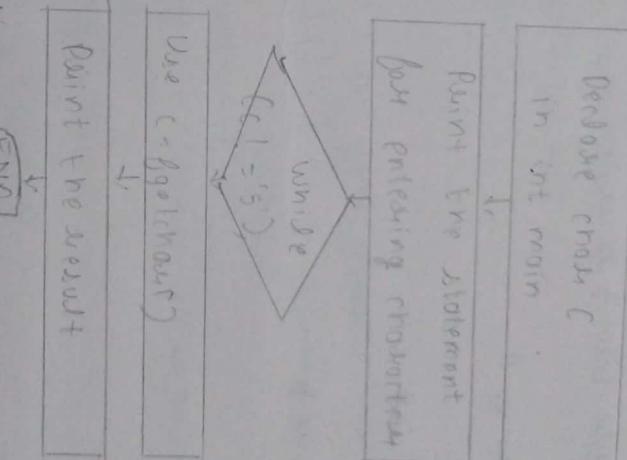
int main()
{
    int num;
    FILE *fptr;
    if ((fptra=fopen ("C:\Programme\text", "w"))!=NULL)
    {
        printf("File Existed opening file\n");
        exit(4);
    }
    fscanf(fptr, "%d", &num);
    pointf ("Value one = %d", num);
    fclose (fptr);
    return 0;
}
```

B3) WAP for `fgetc()`, `fgetchar()`, `fputchar()`.

Algorithm / Description

- `fgetchar()` is a file handling function
- It is used to read a single character from keyboard input.

Flowchart



Output:

Enter some character. Enter \ to exit.

A
Entered character is A.

B
Entered character is B.

C
Entered character is C.

D
Entered character is D.

E
Entered character is E.

Code:

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

`fgetc()` used to read a character from a file
Reads single character at a time.

In a program we use `fgetc()` function

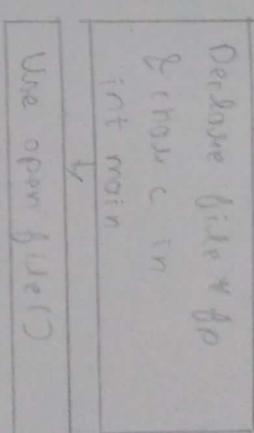
`fgetc(fp)`

where `fp` = file pointer

Note:

```
#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);
    }
}
```

Diagram



Output: moving file test.c in read mode.

Opening the file test.c

Reading the file test.c

Tell, how are you?

Moving the file test.c

painting! Moving file test.c,
close (fp)
return 0;

}