```
#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: %od\n", roll_no);
        printf("student's mobile no: %s\n", mobile_no);
        printf("student's percentage: %f\n", percentage);
        getch();
```


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

Theory:

a] 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 characters & percentage as float.

Step-2: Use printf function to print questions for user in order to give input.

Step3: Use scanf function to read user's input and store in its allocated memory.

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

b] Write a program in C to explain ternary operator.

Algorithm:

Step 1: Declare variables a, b & x as integers

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

Viva

Output:

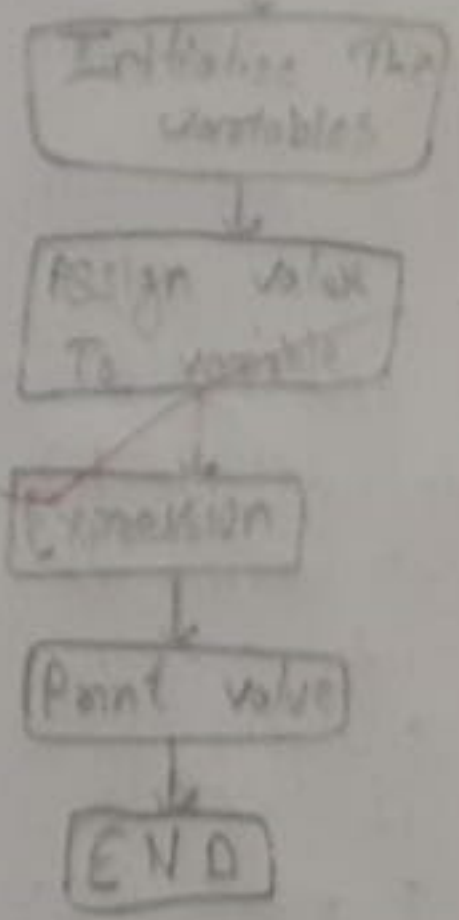
Enter first number: 3
Enter second number: 3
Addition of 3 and 3 is 6
sub. of 3 and 3 is 0
multiplication of 3 and 3 is 9
Division of 3 and 3 is 1

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

Output
15

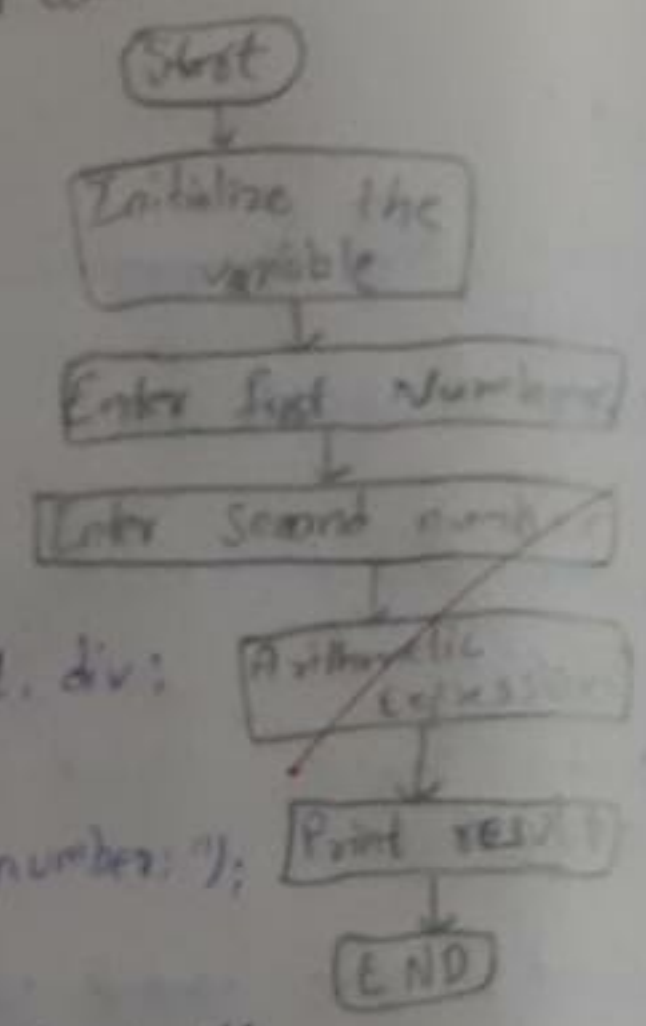
Flowchart



a) Code:

```
// dynamic calculator
#include <stdio.h>
#include <conio.h>
void main()
{
    int num1, num2;
    float add, sub, mult, div;
    clrscr();
    printf("Enter first number: ");
    scanf("%d", &num1);
    printf("Enter second number: \n");
    scanf("%d", &num2);
    add = 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();
```

Flow chart



Practical - 2

Aim: Write a C program on operator and expression

Theory:

a) Write a program to create a dynamic calculator.

Algorithm:

Step 1: Declare a variable name for first & second numbers as integer

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 use expression num1 - num2.

Step 5: Again use expression num1 * num2 if user wishes to multiply the two inputs.

Step 6: Use expression num1 / num2 if user wishes to divide the two inputs.

Step 7: Now use printf function to display output.

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

Output:

```
Enter student's name:
Nitesh
Enter student's roll no:
1885
Enter student's percentage:
59.08%
Enter student's Mobile no:
9999999999
```

Student's name: ~~Ankit~~ Nitesh
Student's roll no.: 1885
Student's mobile no: 9999999999
Student's percentage: 59.08%

:8

Output:

20 is greater than 15 & 12.

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

Nirmodi

Step 3: If condition is true the 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

Algorithm:

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

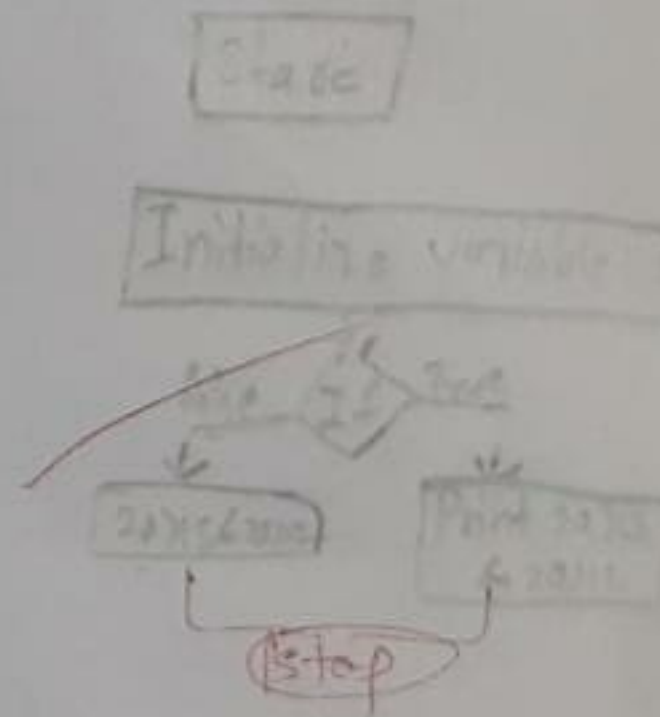
Step 2: Now use nested if logic to compare if given no. is greater or ~~not~~ not.

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


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

Output:

20 is greater than 15



c] code:

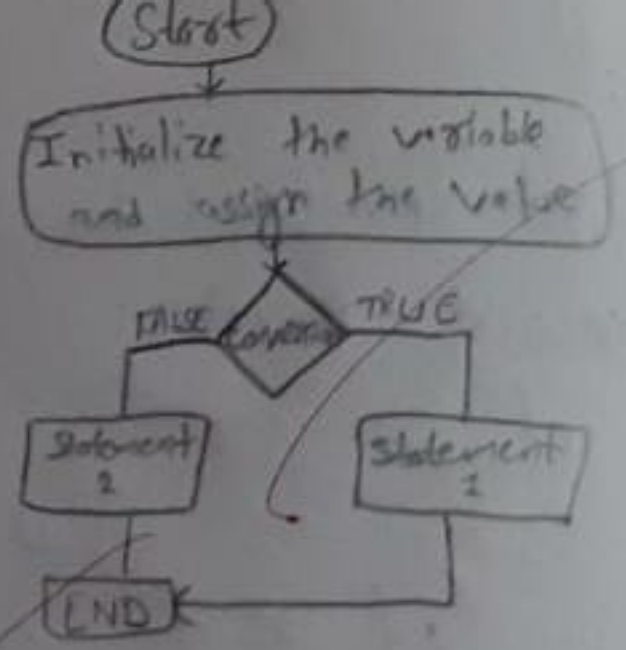
```
// nested if
# include <stdio.h>
# include <conio.h>
void main();
{ int i=20;
clrscr();
if(i<15);
{ if (i<15)
  { printf(" 20 is less than
      15 & 12\n");
  }
}
else:
{ printf(" 20 is greater than
      15 & 12\n");
}
getch();
}
```

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();
}
```

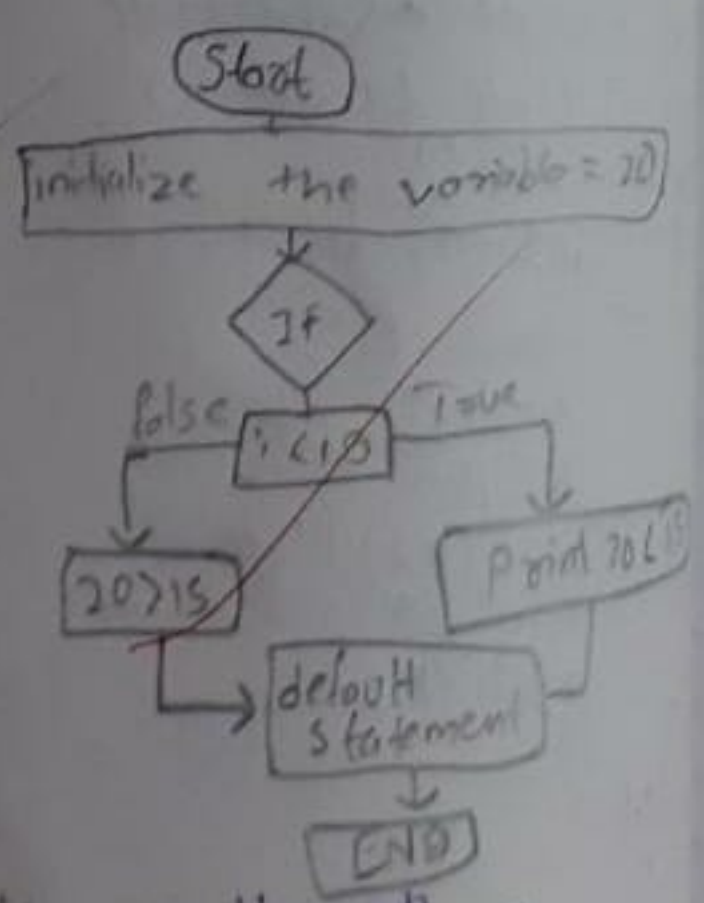
Output

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 ");
    }
```



Aim: 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:

Step1: Declare a variable as integer and assign its value ie. 20

step2: Now to compare the given value 20 is greater than 15 else if statement.

step3: If the condition is true, printf that 20 is less than 15 if condition is false skip the if statement & print 20 I am not in it.

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

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

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

```
#include (conio.h)
#include (stdio.h)
void main()
{
int n1=0, n=2=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

1 1 2 3 5 8 13 21 34 55

Aim - Write a C program of Fibonacci series

Algorithm -

Step 1 - Start the Turbo C

Step 2 - Declare the variable n1, n2, n3, i, number

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

Step 4 - Enter the number terms of fibonacci
series to be printed

Step 5 - Print first 2 terms of series as
$n1=0$ $n_2=1$.

Step 6 - Use the for loop as per following step

$n_3 = n_1 + n_2$;
$n1 = n2$
$n_2 = n3$

Increas

Step 7 - Print the value of number.

Step 8 - END

Conclusion - Thus we have successfully executed fibonacci series on

Practical No. - 4

Aim: To display the prime numbers using for loop.

Algorithm:

Step 1- Initialize a for loop from 1 to 50 & set the count variable zero

Step 2- Nest another loop within the loop step & that goes to 2 to the first variable % 2.

Step 4- Use the if condition statement & check whether the first loop variable % 2 == 0. if true increment the count variable by 1.

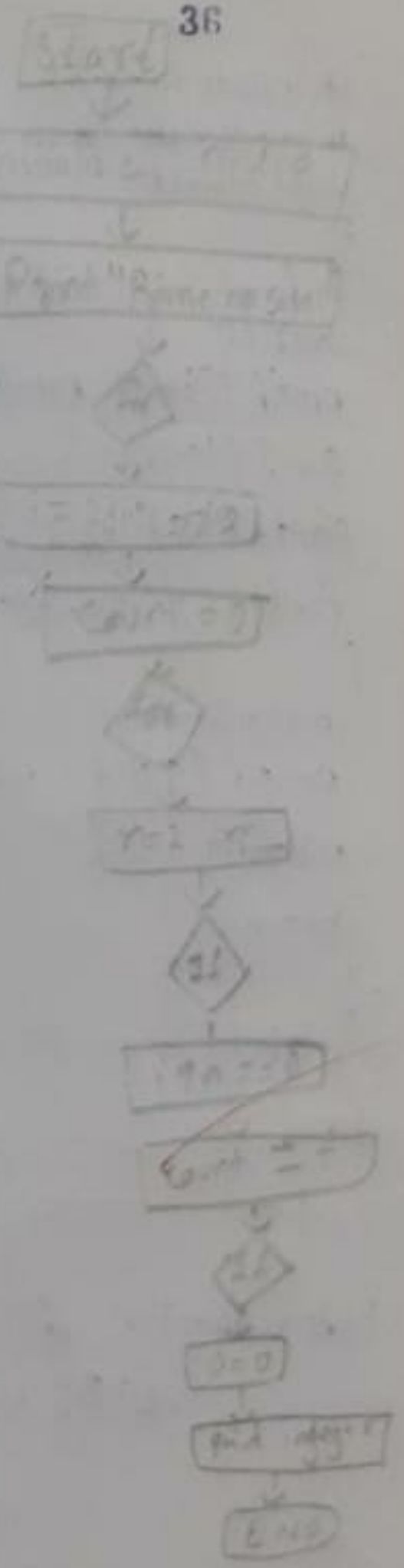
Step 5- Come out of the second loop and check whether the count variable is 0 if true print the number.

Step 6- Terminate the program.


```
#include <conio.h>
#include <stdio.h>
void main()
{
int n,i,a
clrscr();
printf("The prime number 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 number are: 2 17
3 19
5
7



Q.

Write a C program on following expression

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

Algorithm -

Step1- Stat the Turbo C.

Step2- Declare the variable rows i, j, number

Step3- Display the number of rows.

Step4- Enter the for loop. i= 1; i <=rows, i++

Step5- Create nested for loop j=1, j<=i, ++j

Step6- Display the number as per user enter the sequence form

Step7: Increment number from 1

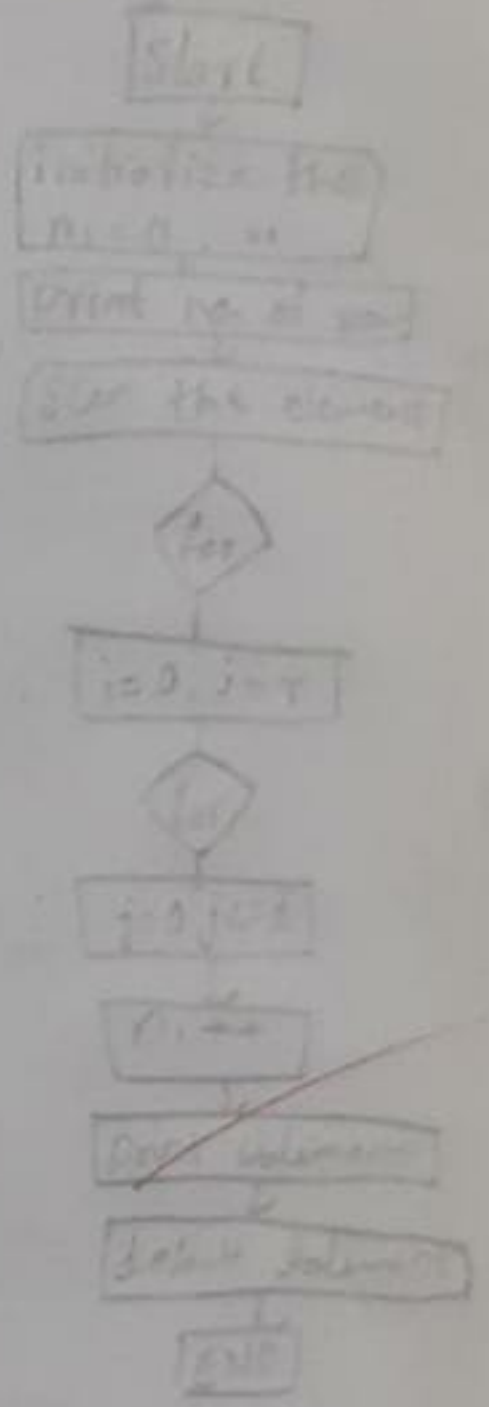
Step8: Display the ~~spac~~

Conclusion: Thus, the we have successfully executed given aprogram as Turbo C using nested for

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n1=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++)
  {
  n1++;
  printf("%d", n1);
  }
  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
```

Source code:

```
#include <stdio.h>
#include <conio.h>
 void main()
{
    int a [100 , i, num ]
    float sum = 0, avg;
    clrscr();
    printf (" Enter no. of elements(in
    scanf(" %d", num; i++)
    for (i= 0; i<num ; i++)
    {
     printf (" \n");
     scanf ("%d", &a[i]);
     sum. += a[i];
    }
    avg = sum /num;
    printf (" \n the sum of no. is %d" sum);
    getch();
```

Conclusion:-

Thus the programs is successfully executed.

Output:


```
Enter the no. of element-to
2
3
6
5
10
1
2
1
7
11
The sum of the no. a is 48.0000
and the avg is 480000.
```

#

printf("%od", array[i]);

}

getch()

}

Output :-

Enter the number of elements: 5

1, 4, 6, 8, 9

Even no: 4

6

8

Odd no.: 1

9

Algorithm:

Step 1 - Start turbo C application.

Step 2 - Initialize the int variable a[100], num, 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

Aim - WAP to print the even and o
numbers in an array

Algorithm -

Step 1 - Create an array taking siz
from user and declare an
array using loop

Step 2 - Display the size of the array

Step 3 - Display the element of arr
entered by user.

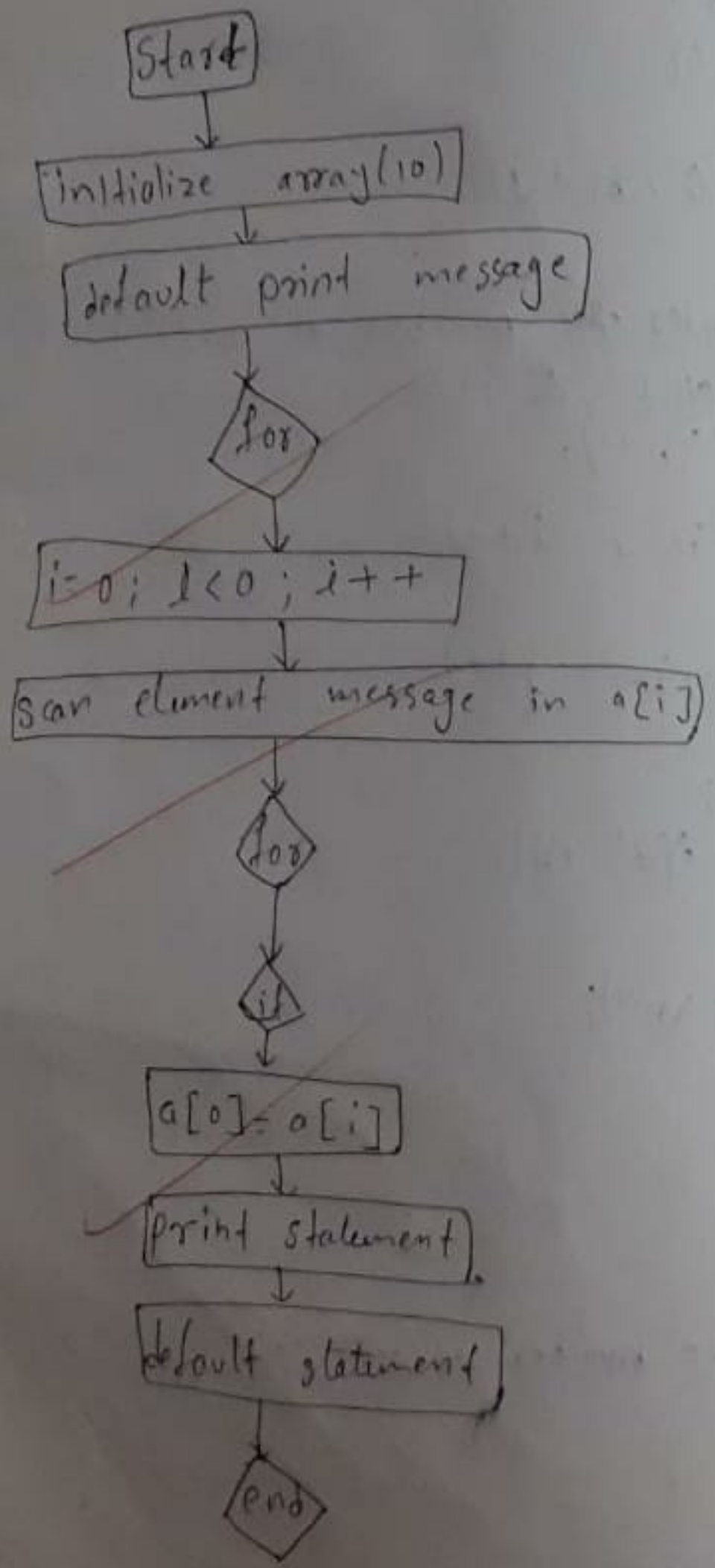
Step 4 - Take the initiator in a for loo
using which all the elemen
in the array

Step 5 - Display even no. from array b
for loop if (array[i]%2==0
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 the C.


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

{
if(array[i]%2!=0)
```



Practical - 5

Aim - C program to find largest array number using array.

Algorithm:

Step 1 - Start turbo c ++ application.

Step 2 - Declare the variable in and integer array a[10]

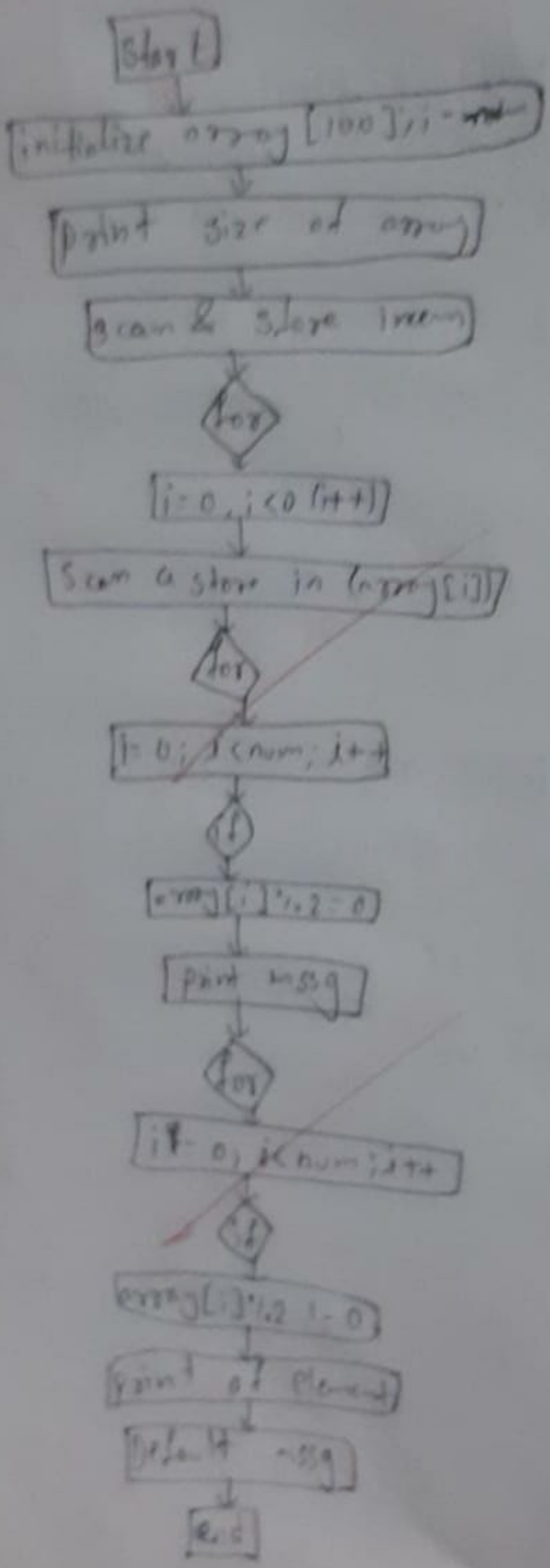
Step 3 - Enter the for loop of i=0, i<10 and use the value of a[i] till i<10. Exit the for loop.

Step 4 - Enter the for loop of i=0, i<10 use if conditional statement to check if a[0] < a[i] if true, put a[0] = a[i]

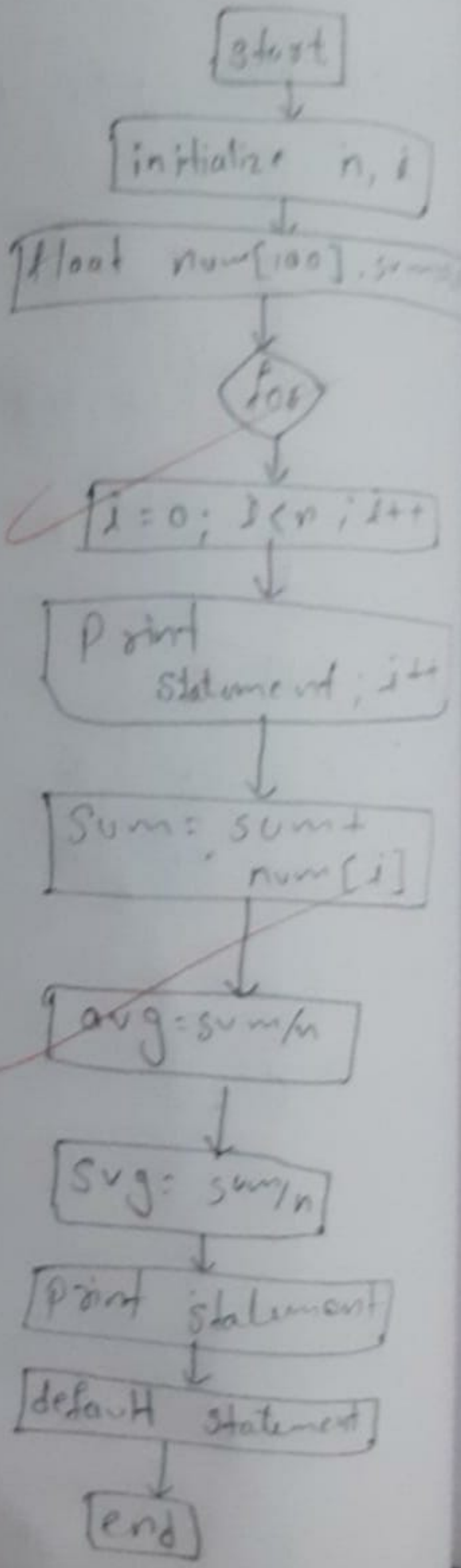
Step 5 - Run the above for loop for i<10, exit the loop.

Step 6 - Terminate the program

Flowchart



Flowchart



8.

Source 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 n, a;
printf (" enter a positive integer ");
scanf ("%d", &n);
a = factorial (n);
printf ("n factorial of %d is %d", n, a)
getch ();
}
```

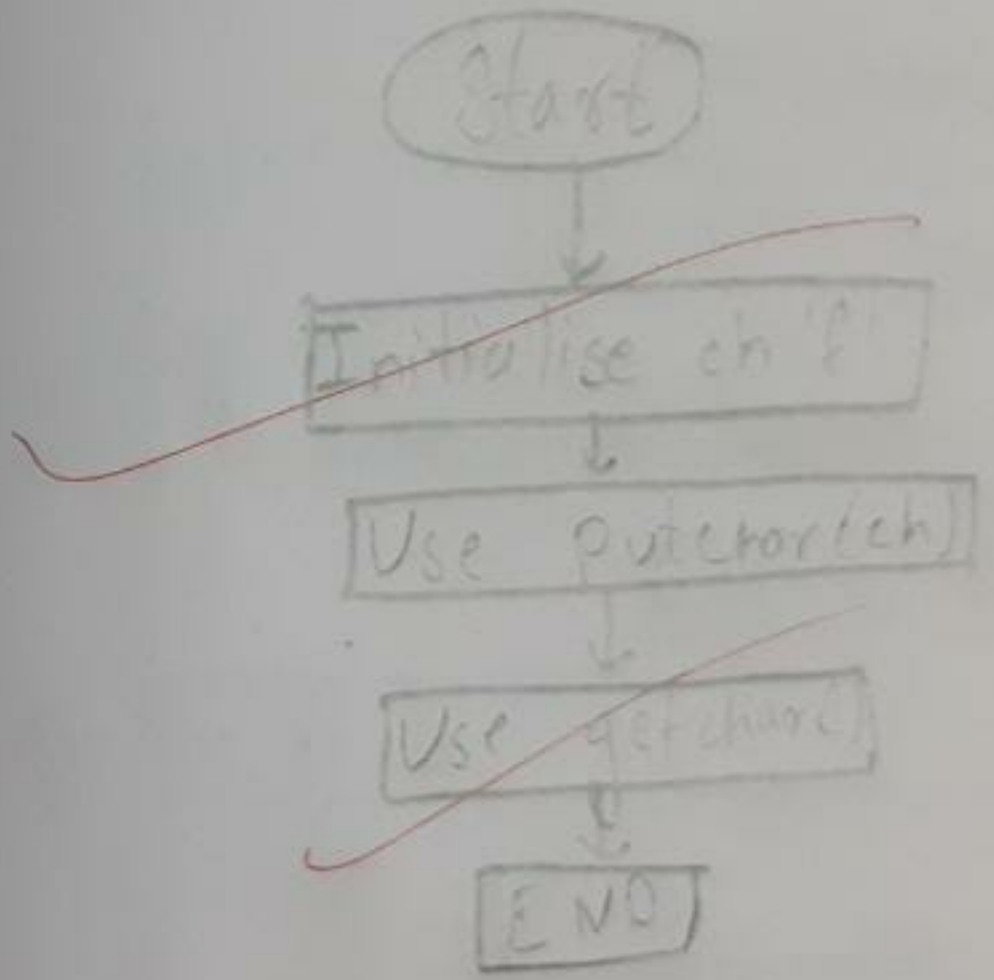
Output -

Enter a positive integer: 5
Factorial of 5 is 120

Source code:

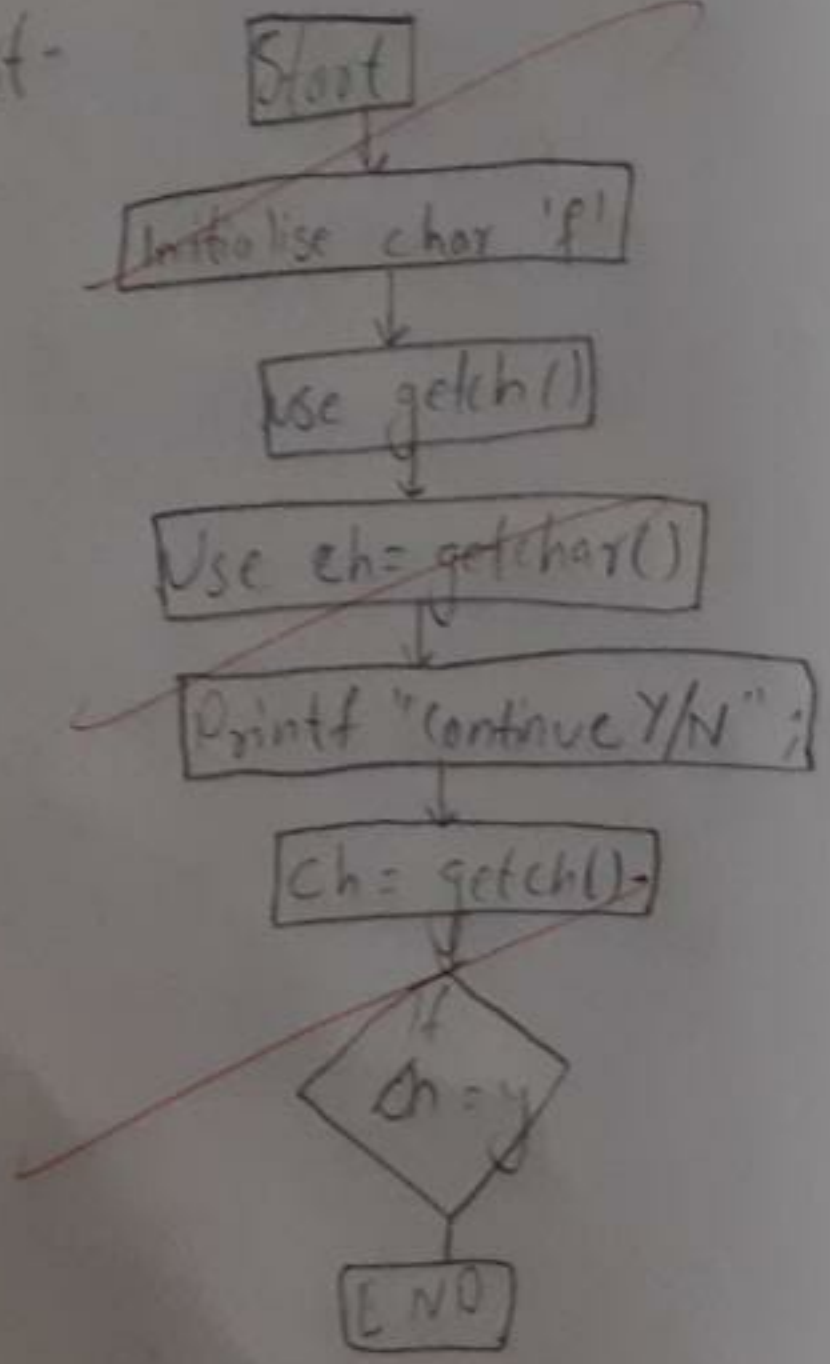
```
# include <stdio.h>
# include <conio.h>
  void main()
{
 char ch;
printf("\n Press any key to continue");
getch();
printf("\n Enter an alphabet");
ch = getch();
printf("\n continue Y/N");
getchar();
}
```

(c) Flow chart:-



Output
Press any key to continue
Enter any alphabet 8
continue Y/N - 8N

Flowchart-



(B) Aim- WAP which shows the use of getch function

Algorithm :-

Step 1 - Start Turbo C application.

Step 2 - Declare char variable ch.

Step 3 - Use print statement of entering any key to continue

Step 4 - Use getch() function

Step 5 : Use print statement for entering alphabet

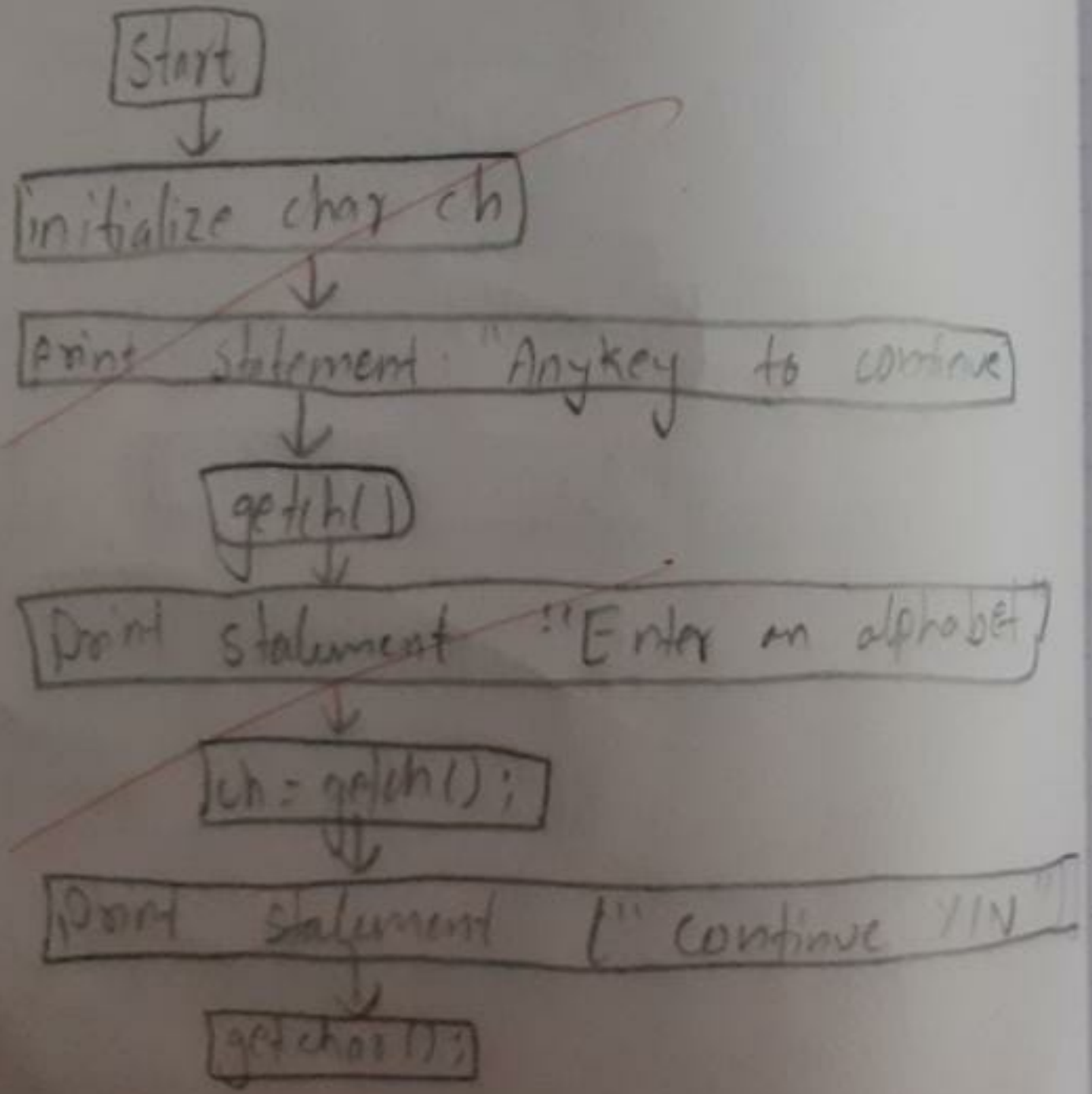
Step 6 - Assign ch to getch() function.

Step 7 - Use print statement for asking to continue the process.

Step 8 - Use getchar() function

Step 9 - Terminate the program.

Flowchart -



(5) Aim - WAP for showing the use of put function.

Algorithm:

Step 1 - Start Turbo C application.

Step 2 = Initialize char ch as b

Step 3 - Use putch function to ch.

Step 4 - Use pitchar function.

Step 5 - Use getch function.

Step 6 - Terminate the program.

Practical - 6

Aim- To find factorial using recursion.

Step1- Start Turbo C application.

Step2- Declare the int variable factorial, n.

Step3- Use if conditional statement and return factorial, and use else statement for returning 1.

Step4- Declare int variable n & 9.

Step5- Use print statement for taking input from user..

Step6- Factorial of n is 9.

Step7- Use default statement

Step 8- Display the output.

Step 9- Terminate the program.

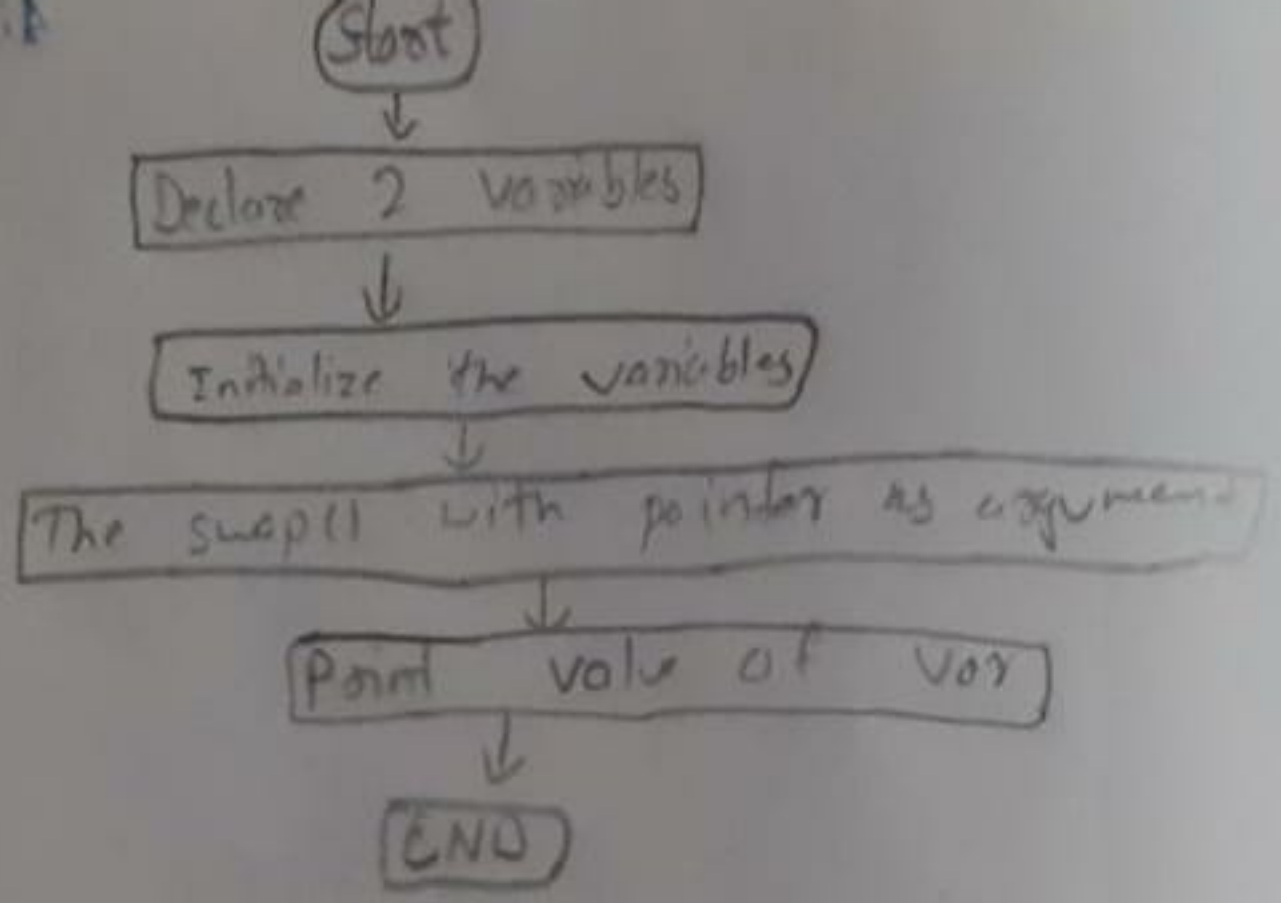
Source code:

```
#include <stdio.h>
#include <conio.h>
    void main()
{
char ch = 'b';
clrscr();
putch(ch);
putchar(ch);
getchar();
}
```

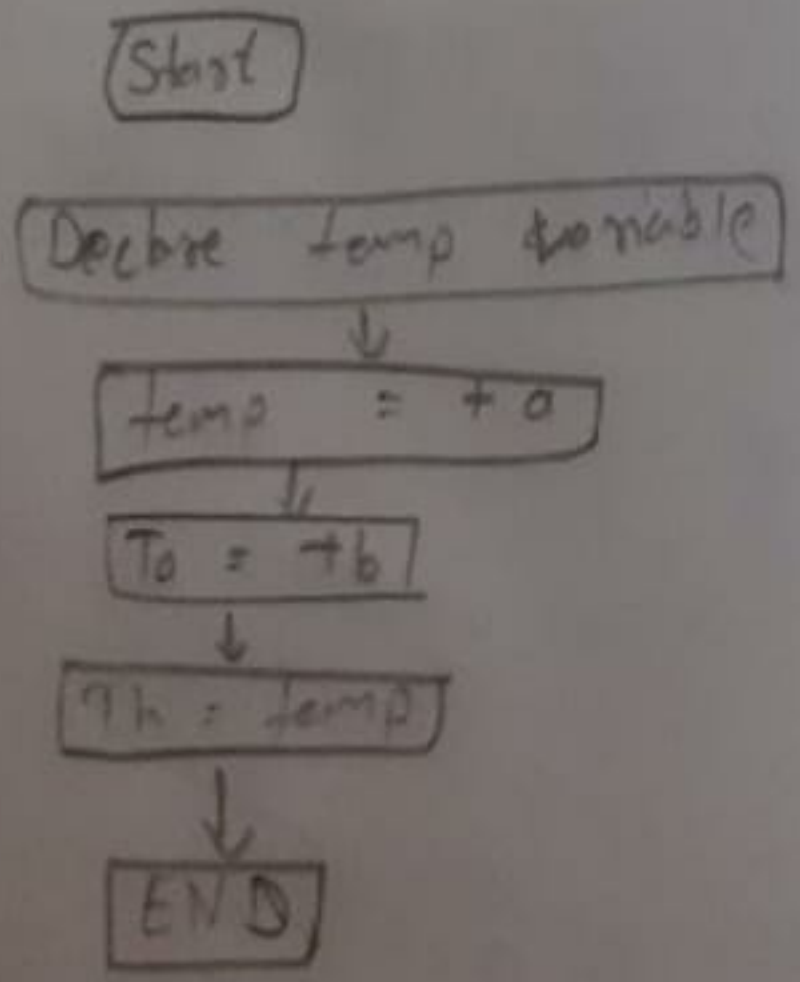
Conclusion:

Thus, we have executed successfully.

Umaadi



Swap -



Practical - 7

Aim -

Algorithm:

Step -1 - Start the turbo C application

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

Step 3: Declare 2 variable and accept thos value from the user print the repective value using printf.

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

Step 5: Print the ~~pepe~~ repective value of variable

Step 6: Use the basic swapping algorithm in the function definition but instead of normal variable use.

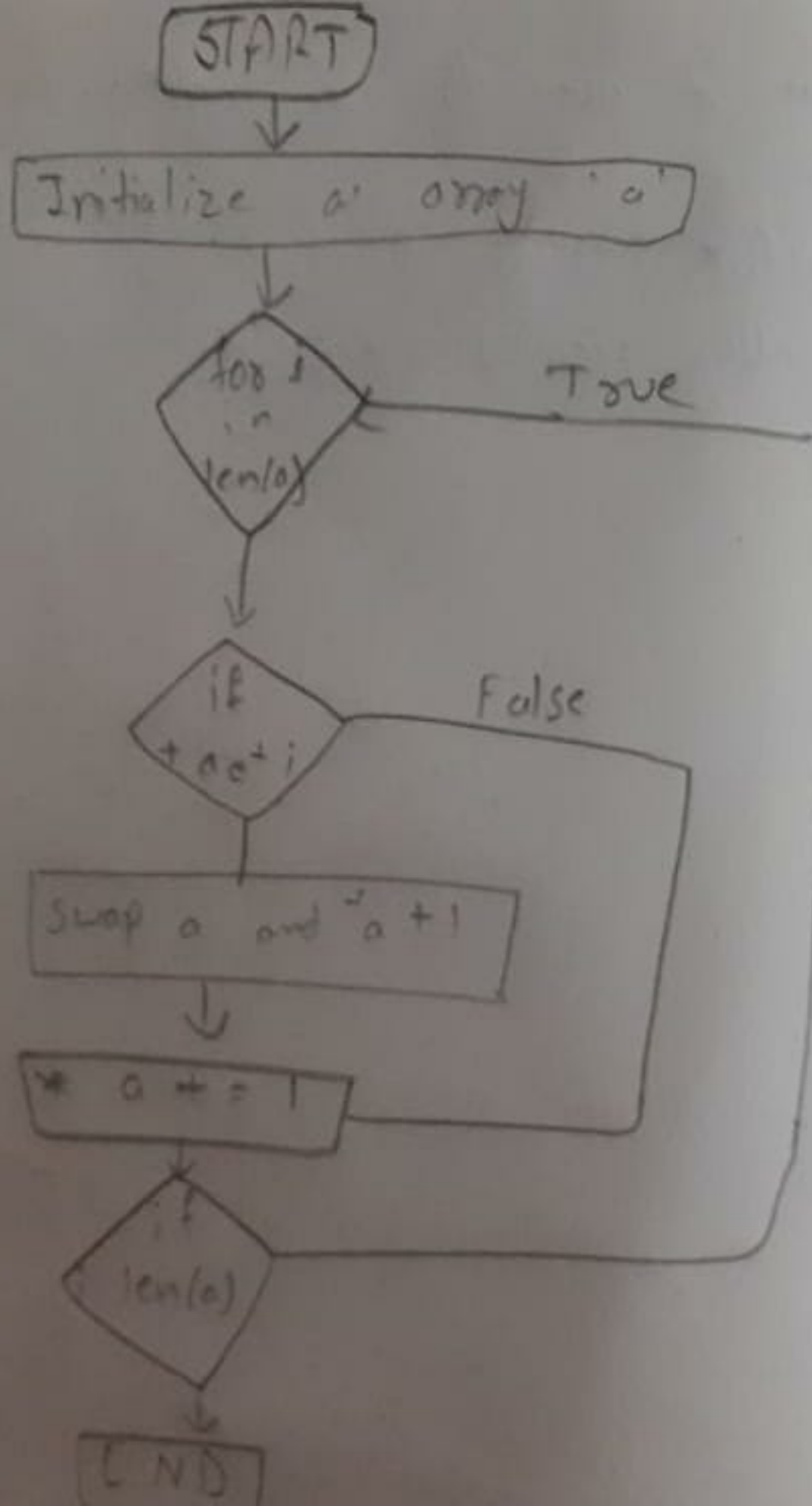
Code:

```
# 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<0-i; j++)
{
   if (*a, *a+1)
    {
     temp = *a+1;
     *a+1 = *a;
     *a = temp;
    }
 }
}
printf("%d is the sorted array", a);
getch()
}
```

Output:

```
Insert element into the array
  2
  5
  9
  8
  6
  4
  3
  1
```

1, 2, 3, 4, 5, 6, 7, 8, 9 is sorted array



(vi) Sorting : of orange using points

Algorithm:

Step 1: Initialize an integer array. temp variable.

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

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

Step 4: Print the swapped array.

Step 5: Terminate the program.

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("%od %od,&n&m);
  printf("The value before swapping as %d
      & %d respectively",x,y);
  getch();
}
  void main(int *m, int(*n)
  {
    temp = *m;
    *m = *n;
    *n = temp;
  }
```

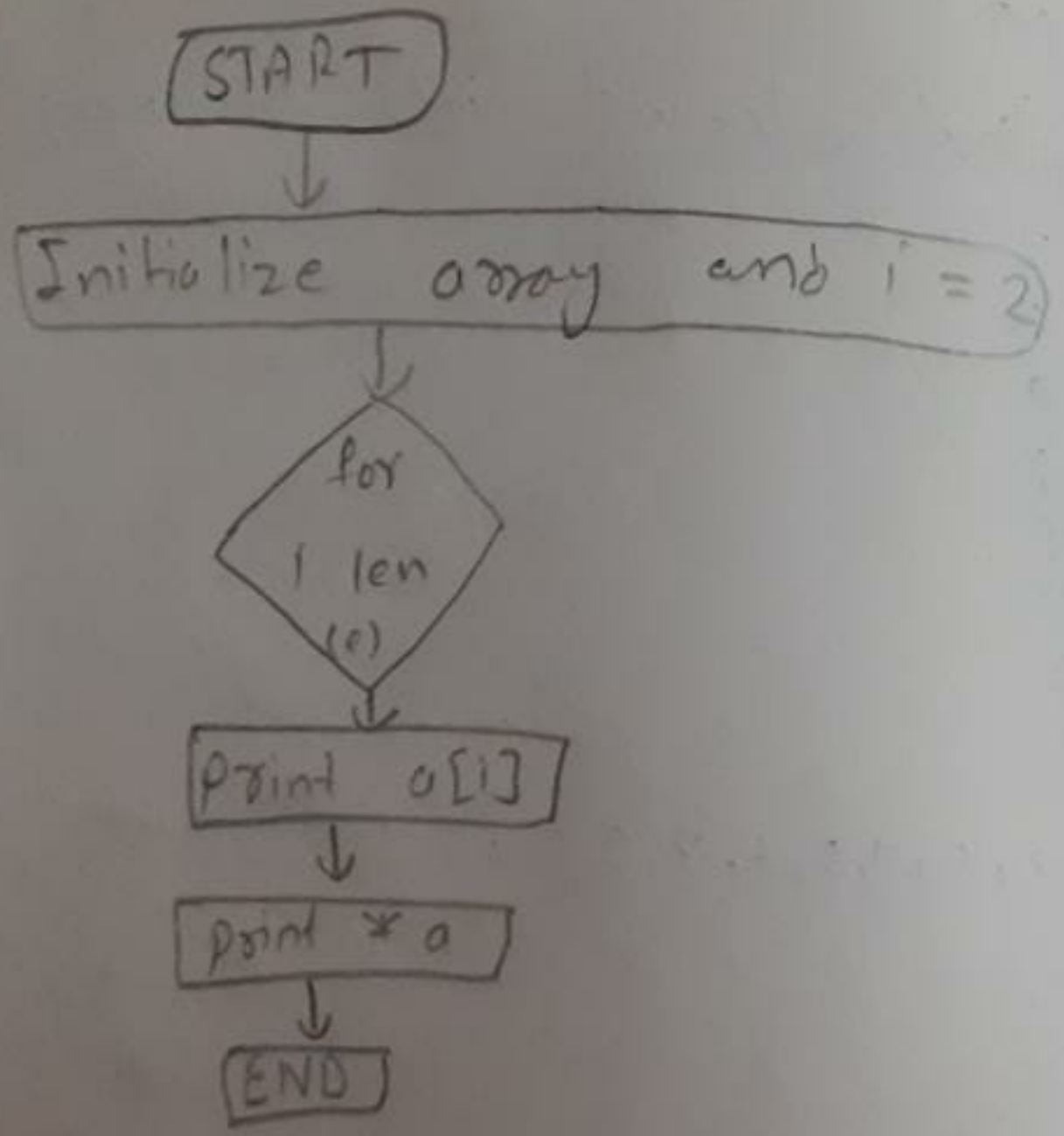
Output :

5n

Enter the 2 no. to be swapped. 12
24

The no. before swapping are 12 & 24

The no. after swapping are 24 & 12



(11) Write a program to find one dimensional arrays using pointer.

Algorithm:

Step 1: Start the turbo C ++ 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:

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

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

Output:

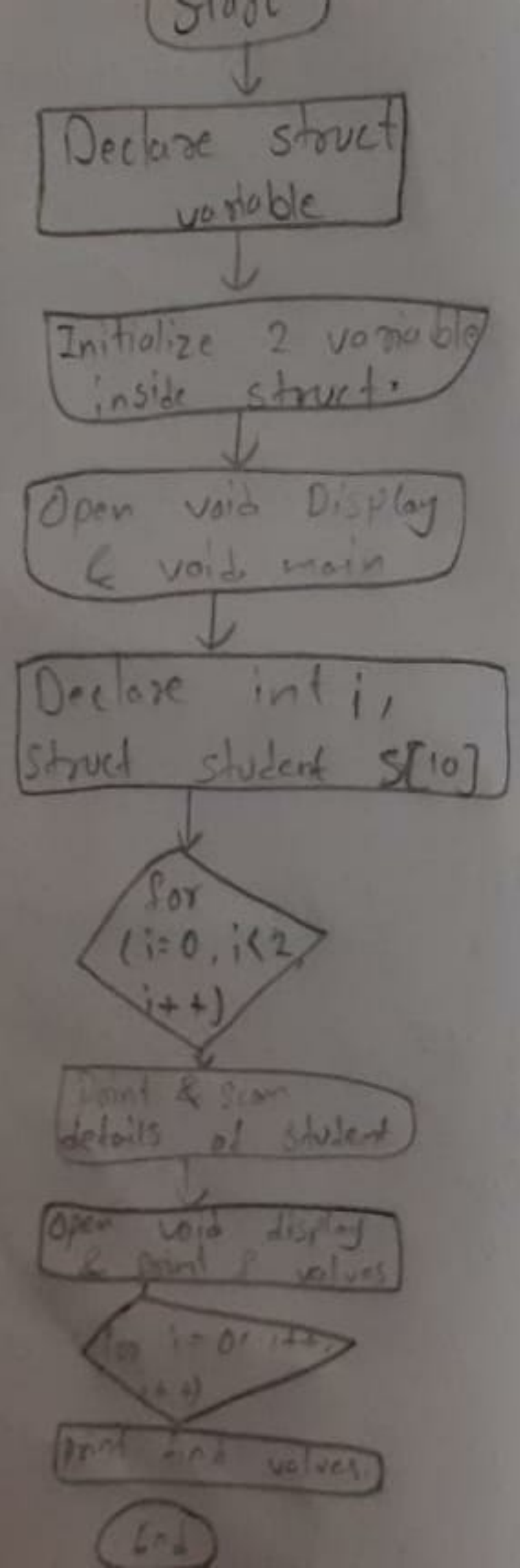
The address of a[0] = 65516
The value of a[0] = 7

The address of a[1] = 65518
The value of a[1] = 5

The address of a[2] = 65520
The value of a[2] = 4

The address of a[3] = 4
The value of a[3] = 4

The address of a[4] = 65524
The value of a[4] = 2



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++)
{
 print ("\n Enter roll and name")
 Scanf ("%d %s", &s[i] roll, s[i] name];
}
 display (s),
 getch();
}
void display (struct student [10])
{
 int i;
 print ("\n * * * * * \n")
 for (i=0, i<2; i++)
printf ("\n roll = %d \t Name = %s, s1[i]. roll. s[i] name);
```

Aim: Create a simple structure that holds following variables id, CGPA, Name.

Algorithm:

Step 1: Start the turbo C application.

Step 2: Declare the ~~stat~~ structured variable as 'struct-student'

Step 3: Initialize the structure student with 3 more variable inside it as 'int-id' 'float (GPA' char-name [10].

Step 4: Now inside void student.

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

Step 6: Terminate the program.

6

3] Aim: UAC P which will demonstrate use of structure and function.

Algorithm:

Step 1: Start the Turbo C Application

Step 3: Initialize the struct student with two 00 integer variables (int roll, char name[10]).

Step 2: Declare the structure variable as struct student.

Step 4: Now Inside void main display and void main declare int i, struct student s[10]

Step 5: Print the details of the student.

Step 6: Open void display again and print the values using condition & print.

Step 7: Terminate the program

5C

Output: (4)

Enter id CGPA and name of student

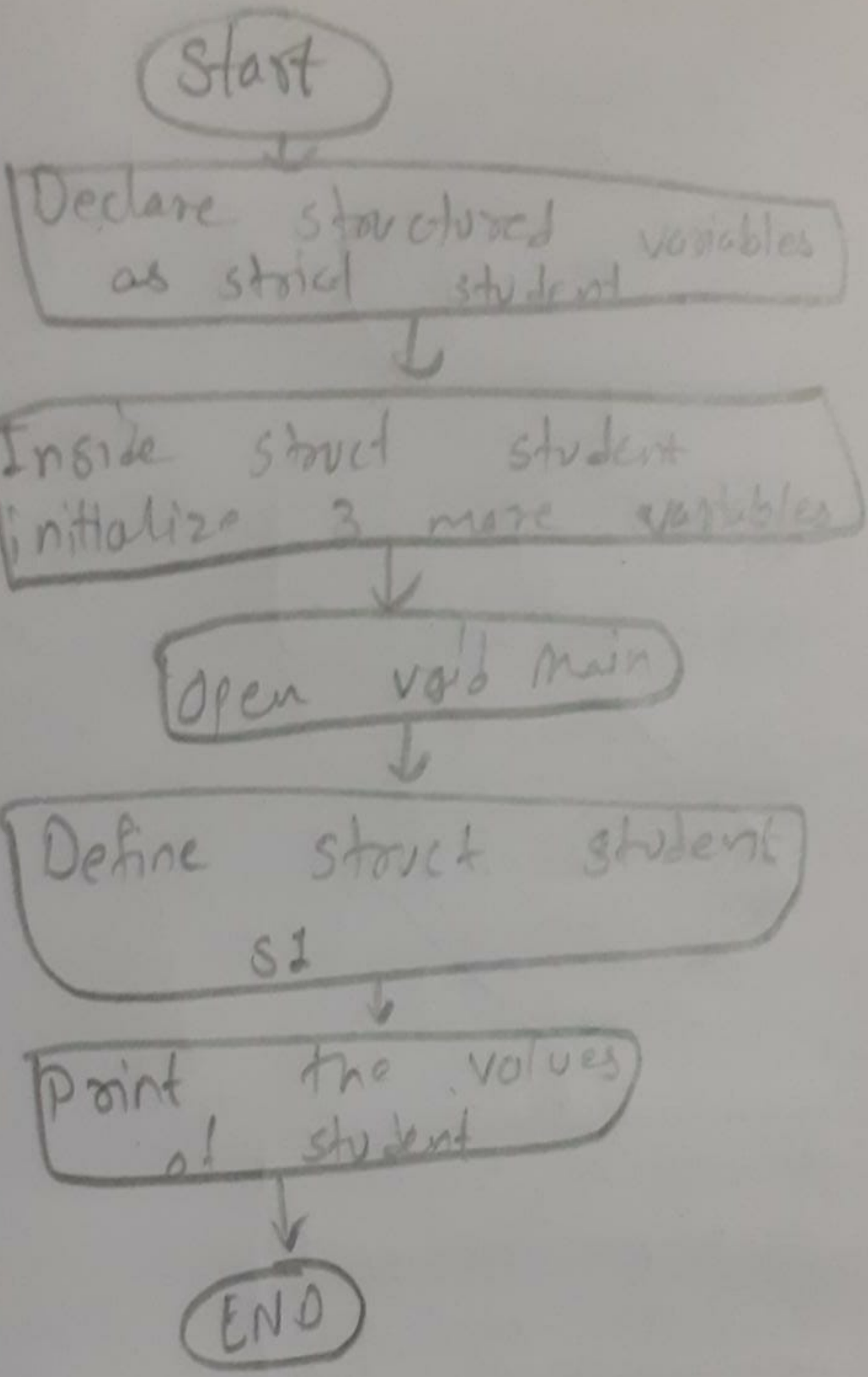
id = 1

CGPA = 6.33

Name - Nitesh

Source Code:

```
struct student
{
int id;
float CGPA
char name[10];
};
void name()
}
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);
}
```

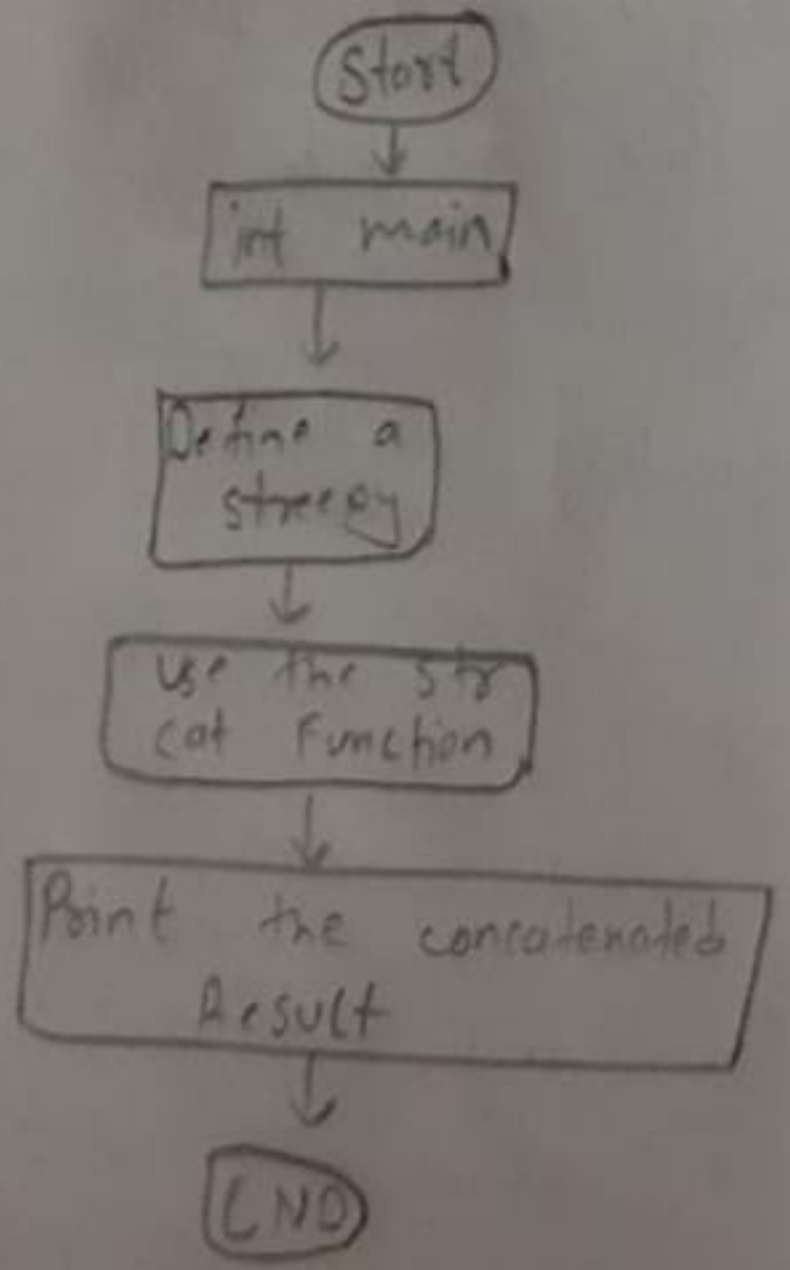

Start
Declare struct variables as struct student
Initialize 3 more variables
Open void main
Define struct student s1
Print the values of student
END

Code:

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

int main (int , argc, const char * argv [])
{
  char example [100];
  strcpy (example "Rahul");
};
strcat (example, " is over 18 ");
strcat (example, " year old ");

printf (" %s\n ", example);
return 0;
}
```

output: Rahul is over 18 year old.

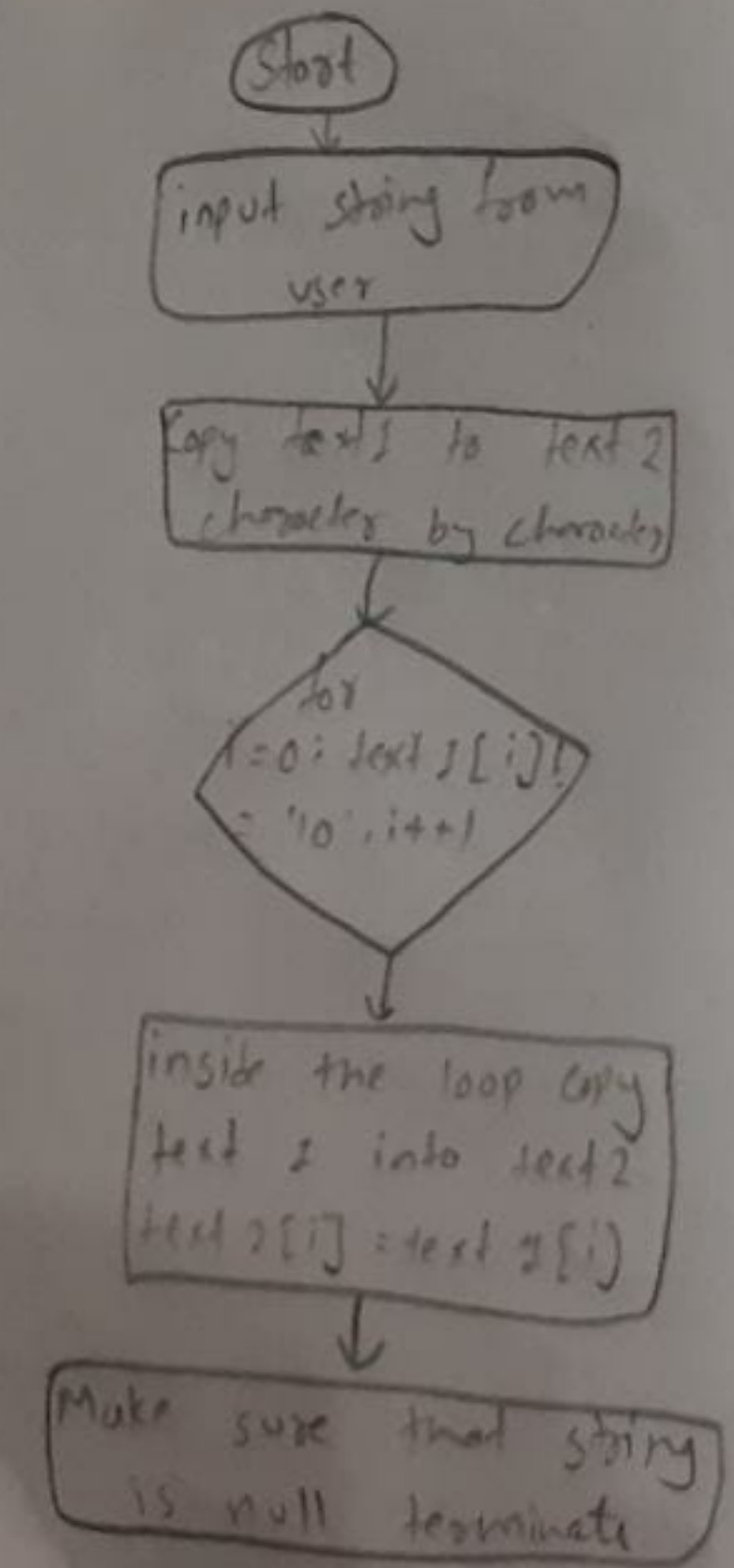
8.



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 argument 1) dest
2) src

The strcat function return a pointer (where the resulting concatenated string resides):



Practical - 9

Aim - WAP to copy one string into another string

Algorithm:

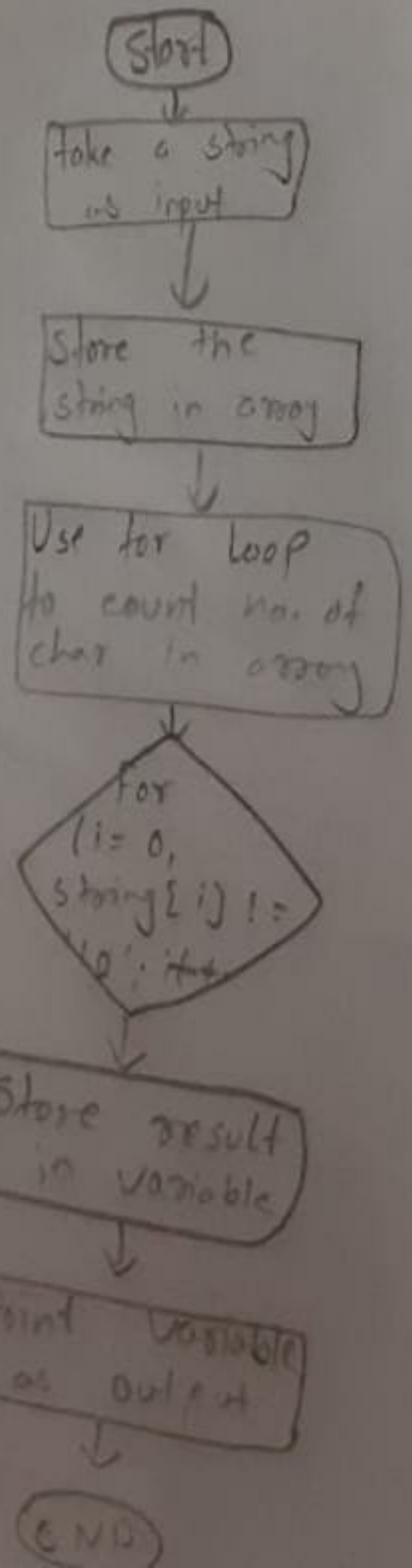
Step 1: Input string from user and store it to some variable say text1.

Step2 : Declare another variable to store copy of first string in text 2

Step3: Run a loop from 0 to end of string 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 with NULL character i.e. text2[i] = '\0';



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

Algorithm:

Take a string as input and store it in the array

Using for loop count the number characters in array

Print the variable as out.

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
            characters in");
    printf(" so the len of : %s = %d \n",
     string, length
```

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 = 18.
```

Code:

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

int main()
{
char text 1[MAX_SIZE];
char text 2[MAX_SIZE];
int ()
printf (" Enter any string:")
gets ( text 1);
for (i=0; text 1[i] != '\0'; i++).
{
    text 2[i] = text 1[i];
}
 text 2[i] = '\0';
printf (" First string = %s\n" text 1)
printf (" First string = %s\n", text 2)
printf (" character copied = %d\n"; i)

  return 0;
}
```

Output:

Enter any string. There are 7 days in a week

First string: there are 7 day in a week

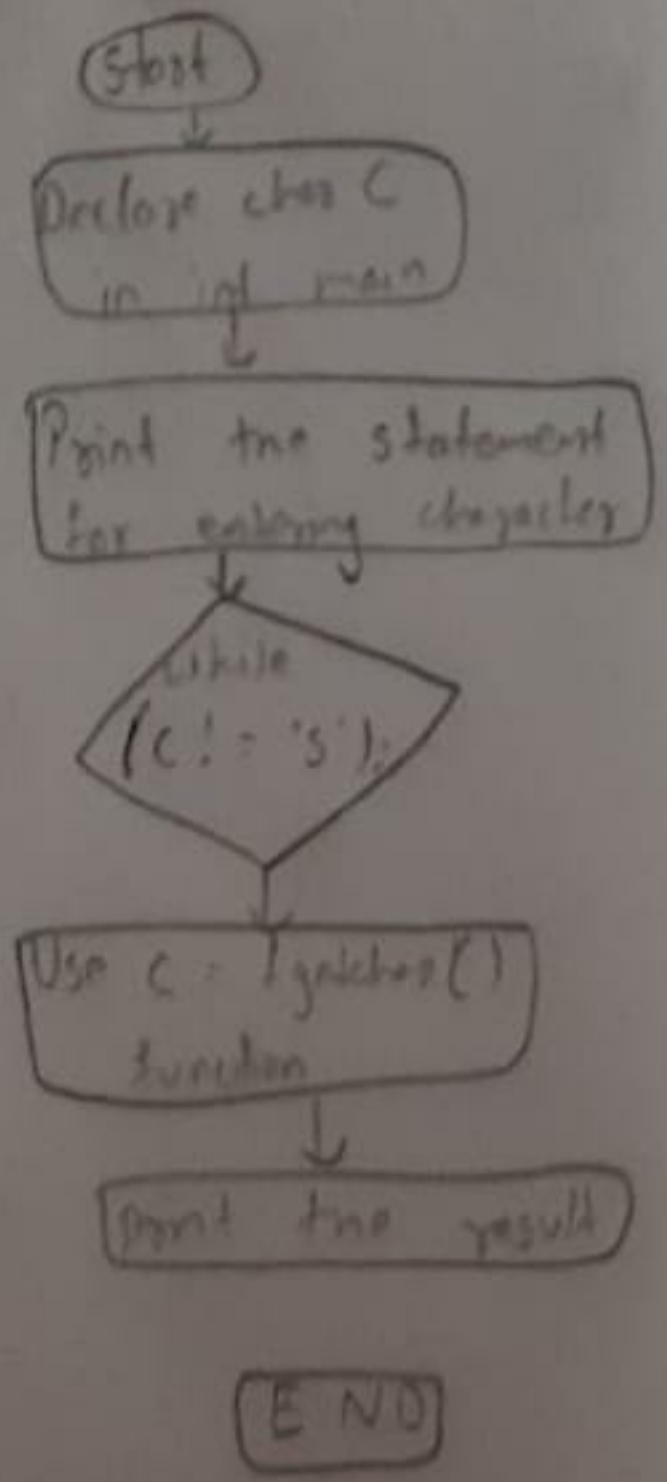
First string copy: There are 7 days in a week

Total character copied: 26.

Aim : WAP for fgetc(), fgetch(), & ftrace() function

Algorithm & Description.

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



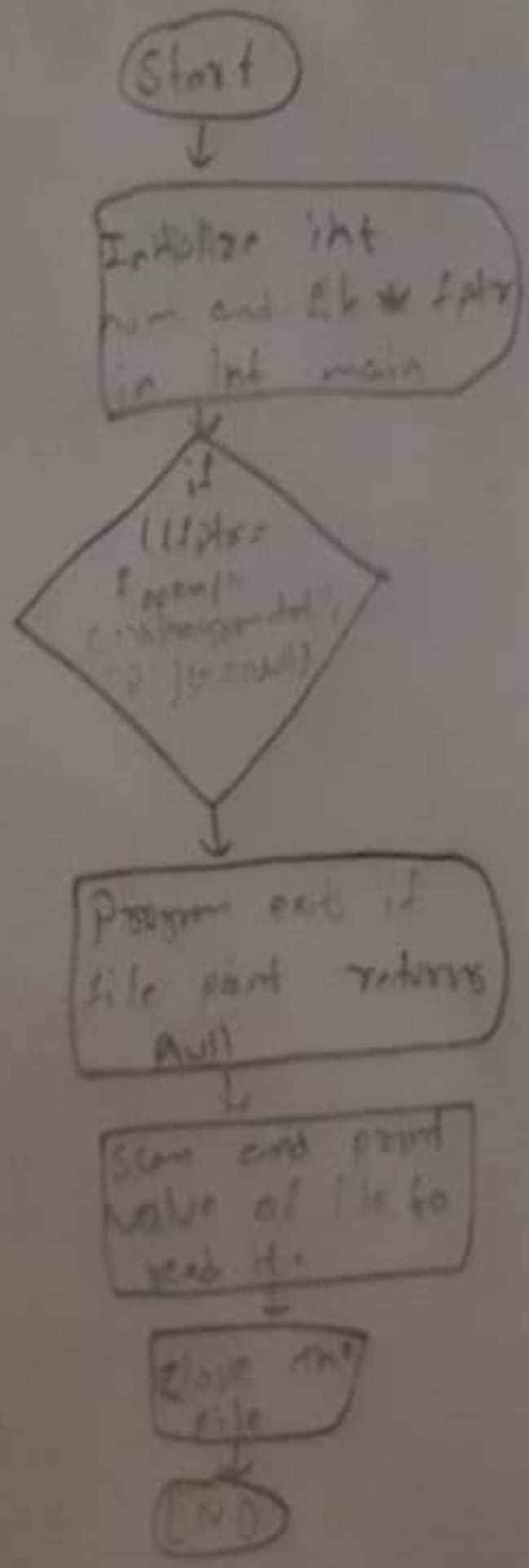
code for reading from a text file opening it.
Text file is 6.txt and its contents
are 87
88
89
90

```
# include <stdio.h>
# include <stdio.h>
int main()
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

10. Read from text file and close it.



Practical 10.

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

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

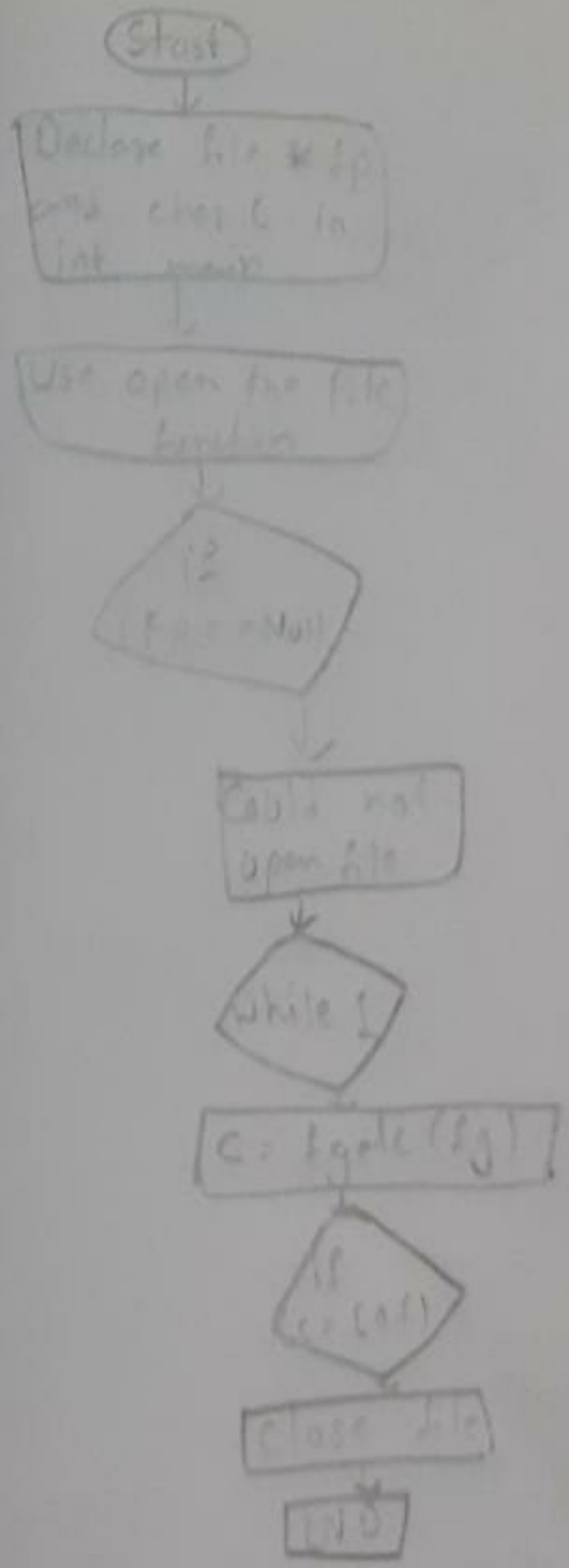
fread() → Read a Record from a file

fclose() → Closes a file.

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;
}
```

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
Read single character at a time.
In a program we use fgetc() function
Where
fp = file pointer

Code:

```
include <stdio.h>
int main()
{
file *fp;
ch=0;
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("Could not open file test.c");
exit(0)
{
c = fgetc(fp);
if (c == EOF)
break
printf("%c",c);
}
```

```
printf("closing file test.c");
fclose(fp)
return 0;
}
```