

## Practical NO: 1

Aim: Write a C program to understand the basic data types.

Algorithm:

step1: Declare a variable integer to store roll numbers, float variables to store percentage and two strings array for storing name and mobile number.

step2: Print using printf method to ask value and use scanf() to get the required value.

step3: After receiving the values print them on by using the printf() statement.

Conclusion:

Thus the Integer, character and float data type been studied.

Notes

8890 880

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

```
void main ()
```

```
{  
    int roll ;  
    float percentage ;  
    char name [40] , mob [12] ;  
    clrscr () ;  
    printf (" enter your name ") ;  
    scanf ("% s " , & name ) ;  
    printf (" enter your name ") ;  
    scanf ("% d " , & roll ) ;  
    printf (" Roll no : % d " , roll ) ;  
    scanf ("% . 2 f " , & percentage ) ;  
    printf (" enter your mob no ") ;  
    scanf ("% . 2 f " , & mob ) ;  
    printf (" mob : % . 2 f " , mob ) ;  
    getch () ;  
}
```

Output :

Enter your name :- Dheen

Enter your mob no:  
mob no: 7875645849

Enter your roll number : 188

roll no:- 188

Enter your percentage :- 73.60  
percentage : 73.60

## Practical NO. 2.

Aim: Write a program on operator and expression.

Expression: (Dynamic calculator)

Algorithm:

Step 1: Declare three variables, accept value to store result.

Step 2: Use the scanf() function to accept two values from the user.

Step 3: Various operators and "+", "-" are to perform arithmetic operation on the variable and store the value in the variable and print it.

Conclusion:

The various expression and binary operator have been studied.

```

#include <conio.h>
#include <stdio.h>
void main()
{
    int
    a, b, result;
    clrscr();
    printf("Enter the two value");
    scanf("%d,%d", &a, &b);
    result = a + b;
    printf("a+b=%d", result);
    result = (a / b);
    printf("a/b=%d", result);
    result = a * b;
    printf("\n a*b=%d", result);
    result (%.int(b));
    printf("\n a%.b=%d", result);
    getch();
}

```

Output : Enter the two value : 18, 10

$$a+b=28$$

$$a-b=8$$

$$a/b=1$$

$$a*b=180$$

$$a%.b=1.8$$

## (A) Ternary Operator:

Algorithm:

step 1: Declare 3 integers variables.

step 2: Use the `scanf()` Method to accept the value.

step 3: Use the ternary operator to store the value of the greater number in the third variable.

step 4: print the greater number.

Conclusion:

The ternary operator work like a conditional 'if else' and store the result and value in a derived value.

Model:

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

```
{  
    int a, b, x;  
    point ("Enter any two numbers");  
    scanf ("%d %d", &a, &b);  
    x = a > b ? a : b;  
    point ("Enter the greater number is");  
    getch();  
}
```

Output :

Enter any two numbers 17 10  
~~The greater number is 17.~~

## Practical NO. 3.

Aim:

(i) If statement  
Algorithm:

Step 1: Take an Integer input after declare method.

Step 2: If the given integer is greater than 15, print that it is greater than 15. During the if conditional statement

Step 3: Also outside of the "if" block print a statement that if is outside the scope of if statement in else

Conclusion:

Then if statement execution a block of code only if a certain contribution in net

```

#include < stdio.h >
void main ()
{
    int a;
    clrscr ();
    printf ("Enter an integer ");
    scanf ("%d", &a);
    if (a > 15)
    {
        printf ("Entered number is greater than 15");
    }
    else
        printf ("This block is out of the if statement ");
    getch ();
}

```

~~The entered integer is greater than 15  
thus block is out of the if statement.~~

Output : Enter the integer : 16

(ii) If - else

Algorithm :

Step 1 : Initiate an integer variable then use `scanf()` method in the input from the user.

Step 2 : Use the `if` statement to check whether the number is less than 20 and print a `else` statement after doing.

Step 3 : Terminate the program.

Conclusion :

The `else` part is executed if the condition of `if` statement is not met.

```

#include < stdio.h>
#include < conio.h>
void main()
{
    int a
    clrscr();
    printf("Enter an integer");
    scanf("%d", &a);
    if (a < 20)
    {
        printf("The entered no is less than 20");
    }
    else
    {
        printf("The no is less than 20");
    }
    getch();
}

```

Output: Enter an Integer: 16  
 The entered no. is less than 20.

## (iii) Nested If

Algorithm:

Step 1: Declare an integer than use the `clrscr()` method to clear the console.

Step 2: Use the if conditional statement to check whether the given number is greater than or equal to 50 use another if statement variable inside the block to check whether the integer is equal to 50 the else statement to print the response that the no is greater than 50.

Step 3: Use the else statement to print that the number is smaller than 50.

Conclusion:

The multiple if statement are nested together.

Nested

```
#include <conio.h>
#include <stdio.h>
void main ()
{
    int a;
    clrscr();
    printf ("enter an integer");
    scanf ("%d", &a);
    if (a >= 50)
    {
        if (a == 50)
        {
            printf ("no is 50");
        }
        else
        {
            printf ("The no is greater than 50");
        }
    }
    else
    {
        printf ("no. is less than 50");
    }
    getch();
}
```

## Practical No. 4.

Aim: To display the prime numbers using for loop.

Algorithm:

Step 1: Initialise the ~~one~~ variable out of which two are loop variable and one ~~one~~ count variable.

Step 2: Initialise a for loop from 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 ~~2~~.

Step 4: Use the if condition statement to check whether (first loop variable  $\% 2$  <sup>nd</sup> loop variable = 10 ; 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.

Conclusion: Prime numbers was displayed using for loop

```

980
#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 ", i);
        }
    }
    getch();
}

```

Output: The prime numbers are:

$\frac{1}{3}$     13  
 $\frac{3}{7}$     17  
 $\frac{7}{11}$     19  
 $\frac{11}{5}$

Aim: Write a C program on fibonacci.

Step 1: Start the turbo c.

Step 2: Declare the variable  $n_1$ ,  $n_2$ ,  $n_3$ ,  $i$ , number.

Step 3: Initialise the variable  $n_1 = 0$ ,  $n_2 = 1$  and number = 0.

Step 4: Enter the no. of term of fibonacci series to be print.

Step 5: Print first two term of series as  $n_1 = 0$  and  $n_2 = 1$ .

Step 6: Use the for loop as per following

$$\text{Step } n_3 = n_1 + n_2 ;$$

$$n_1 = n_2 ;$$

$$n_2 = n_3 ;$$

Increase the value of i element each time by 1.

Step 7: Print the value.

Step 8: End.

Conclusion: We have successfully executed fibonacci series.

```

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

```

Output: Enter no. of elements :

0  
1  
1  
3  
5  
18  
13

Aim: Write a C program on following Expression.

Step 1: Start the Turbo C program.

Step 2: Declare the variable rows, i, j, number.

Step 3: Display the no. of rows.

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

Step 5: Create Nested for loop as  $j=1, j \leq i, j++$

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

Step 7: Increment no. from 1.

Step 8: Display the space.

Conclusion: Thus we have successfully execute given expression on Turbo C using Nested for loop.

*Chmod*

880

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, j, k, num=1;
    clrscr();
    printf("Enter the number of rows");
    scanf("%d", &i);
    printf("\n");
    for (i=0; i<=k; i++)
    {
        for (j=0; j<=i; j++)
        {
            printf("%d", num);
            num++;
        }
        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
16 17 18 19 20 21
```

Practical No. 5

Aim: C program to find largest among array numbers using array.

Algorithm:

step1: Start turbo c application.

step2: Declare the variable i and integer array a[10]

step3: Enter the for loop at  $i=0$ ,  $i < 10$  and use the value of  $a[i]$  till  $i < 10$ . Exit the for loop.

step4: Enter the for loop at  $i=0$ ,  $i < 10$   
use if conditional statement to check  
if  $a[0] < a[i]$  if true, put  
 $a[0] = a[i]$

step5: Run the above for loop for  $i < 10$ ,  
Exit the loop.

Step 6: Terminate the program.

Conclusion: We have successfully executed program.

```
#include <conio.h>
#include <stdio.h>
void main ()
{
    int a[10], i;
    clrscr();
    printf ("Enter the element of the list");
    for (i=0; i<10; i++)
    {
        scanf ("%d", &a[i]);
    }
    for (i=0; i<10; i++)
    {
        if (a[0] < a[i])
        {
            a[0]= a[i];
        }
    }
    printf ("The largest number is = %d", a[0]);
    getch();
}
```

040

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

Algorithm: Aim: WAP to print the number of odd and even no. In the array.

Algorithm:

1. Create an array taking size from user and define it as element using loop
2. Display the size of array
3. Display the element of array entered by user
4. Take the ~~initiator~~ in a for loop using which all the elements in the array exist.
5. Display even no. from the array from for loop
6. If ( $\text{array}[i] \times 2 = 0$ )  
Display even no. from given array
7. Display the odd no.  
if ( $\text{array}[i] \times 2 \neq 0$ )  
then display odd no.
8. Close.

Q10

Output:

Enter the element of the list.

12  
23  
2  
2  
12  
55  
3  
1  
22  
100

The largest number is 100.

Conclusion: ~~We~~ we have successfully executed program.

Q.

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int a[100], num, i;
    clrscr();
    printf ("Enter no. of elements (<100 : ");
    scanf ("%d", &num);
    for (i=0; i<num; i++)
        scanf ("%d", &a[i]);
    printf ("Even no : ");
    for (i=0; i<num; i++)
        if (a[i] % 2 == 0)
            printf ("%d", a[i]);
    for (i=0; i<num; i++)
        if (a[i] % 2 != 0)
            printf ("%d", a[i]);
    getch();
}
```

output:

Enter the no. of elements : 10

042

6  
4  
3  
2  
1  
21  
8  
9  
7  
7

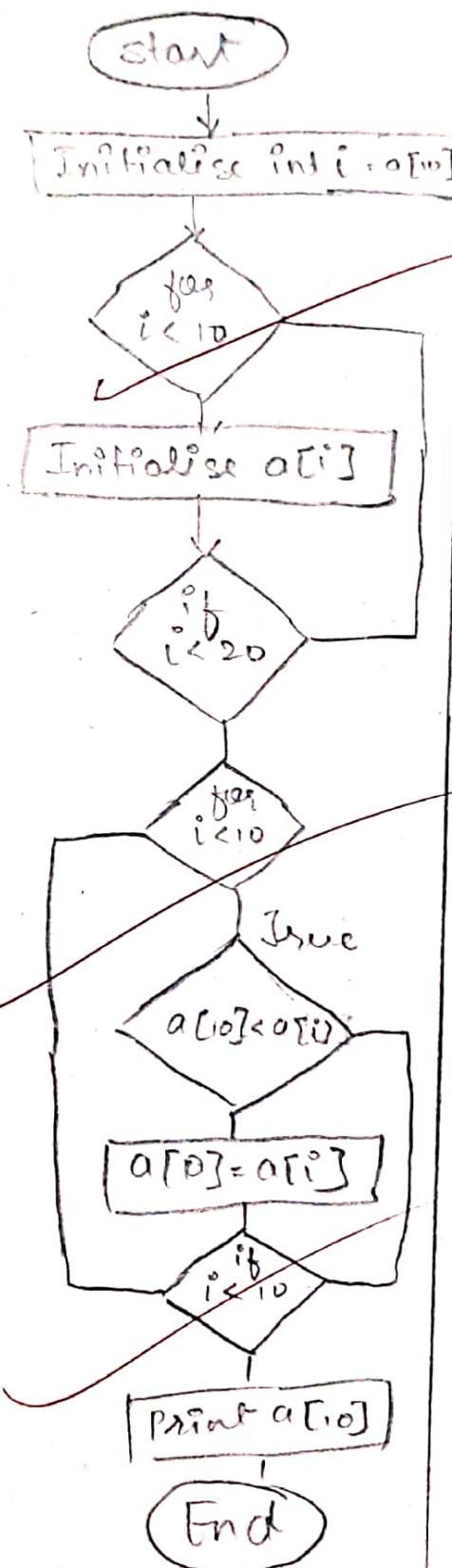
Even no:

6  
4  
2  
8

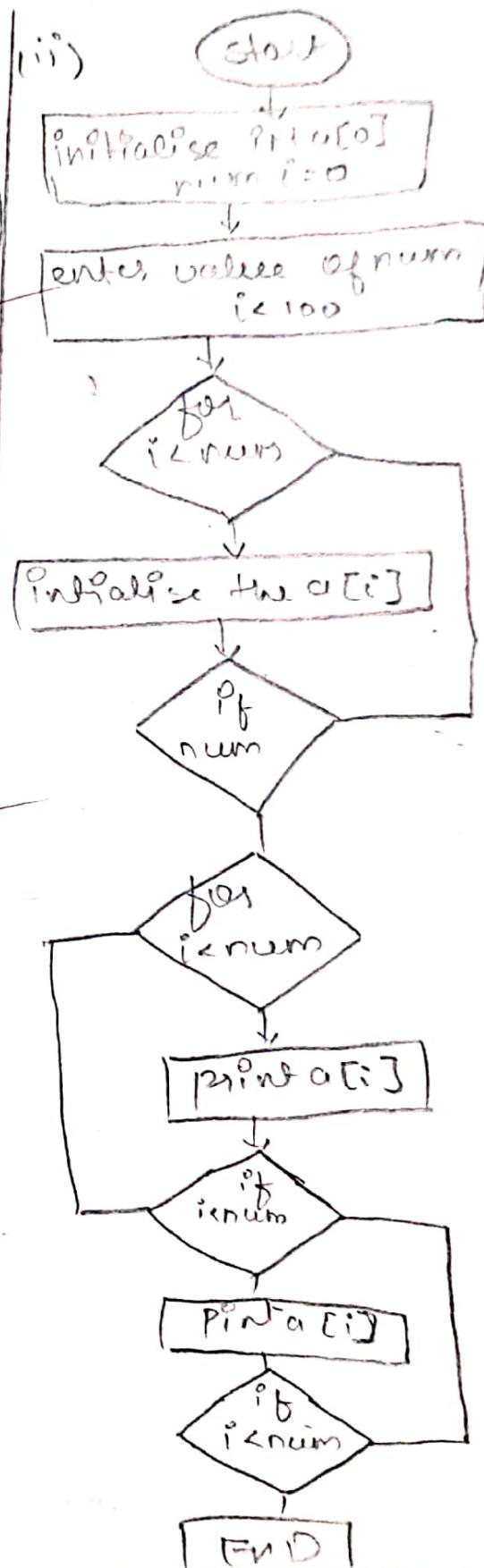
Odd no:

3  
1  
21  
9  
7  
7

(i)



(ii)



## (iii) Algorithm:

- step1: Start turbo C application.
- step2: Initialise the int variable  $a[100]$  num1 and float variable sum=0 and avg.
- ~~step3: Start the user for the length of array ( $<100$ )~~
- step3: Start the user for the length of array ( $<100$ ) and store the value in num using sum & carry.
- ~~step4: Put avg = sum/num~~
- ~~step5: Print the value of sum & average.~~
- step6: Terminate the program.

SSAO

start

Initialize int a[100] num, i = 0, float sum = 0, avg



Initialise the value of num

for  
num > i

Initialise a[i]



Sum = sum + a[i]

if  
i < num

avg = sum / num

print avg and sum

END

```

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

```

Conclusion: The program code we have executed ~~prog~~ have been ~~not~~ successful.

*Ansari*

output: enter the no. of elements.

041

2  
3  
6  
5  
10  
1  
2  
1  
7  
11

The sum of number is 48.0000  
and the avg is 4.80000

Practical No.6

Aim: Factorial of a number using recursion.

Algo:

step 1: Start the C program application.

step 2: Declare a function prototype entering the main () function.

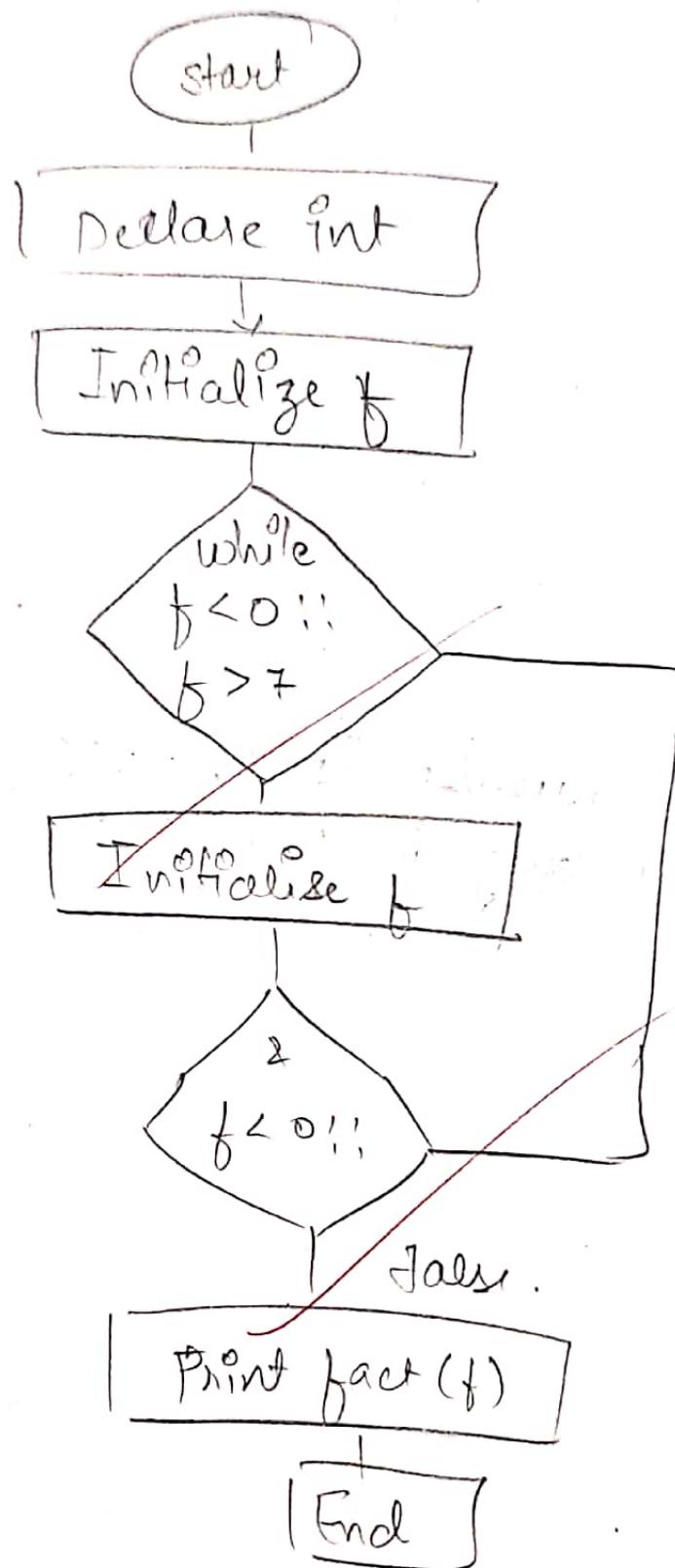
step 3: Declare a variable 'f' in the main () function

step 4: Use the while function to ensure that entered value is in the range 0 to 7.

step 5: Print the value entered by the function in step 4

step 6: Terminate the program.

Q10.



```

#include <stdio.h>
#include <conio.h>
void main()
{
    int f;
    clrscr();
    printf("enter the number to find the
           factorial of (0 to 7)");
    scanf("%d", &f);
    while (f<0 || f>7)
    {
        printf("y.d", fact(f));
        scanf("y.d", &f);
    }
    printf("y.d", fact(f));
    getch();
}

int fact(int n)
{
    if (n>=1)
    {
        return n * fact(n-1);
    }
    else
    {
        return 1;
    }
}

```

Output:

~~enter the number  
of (0 to 7) : 6~~

~~720~~

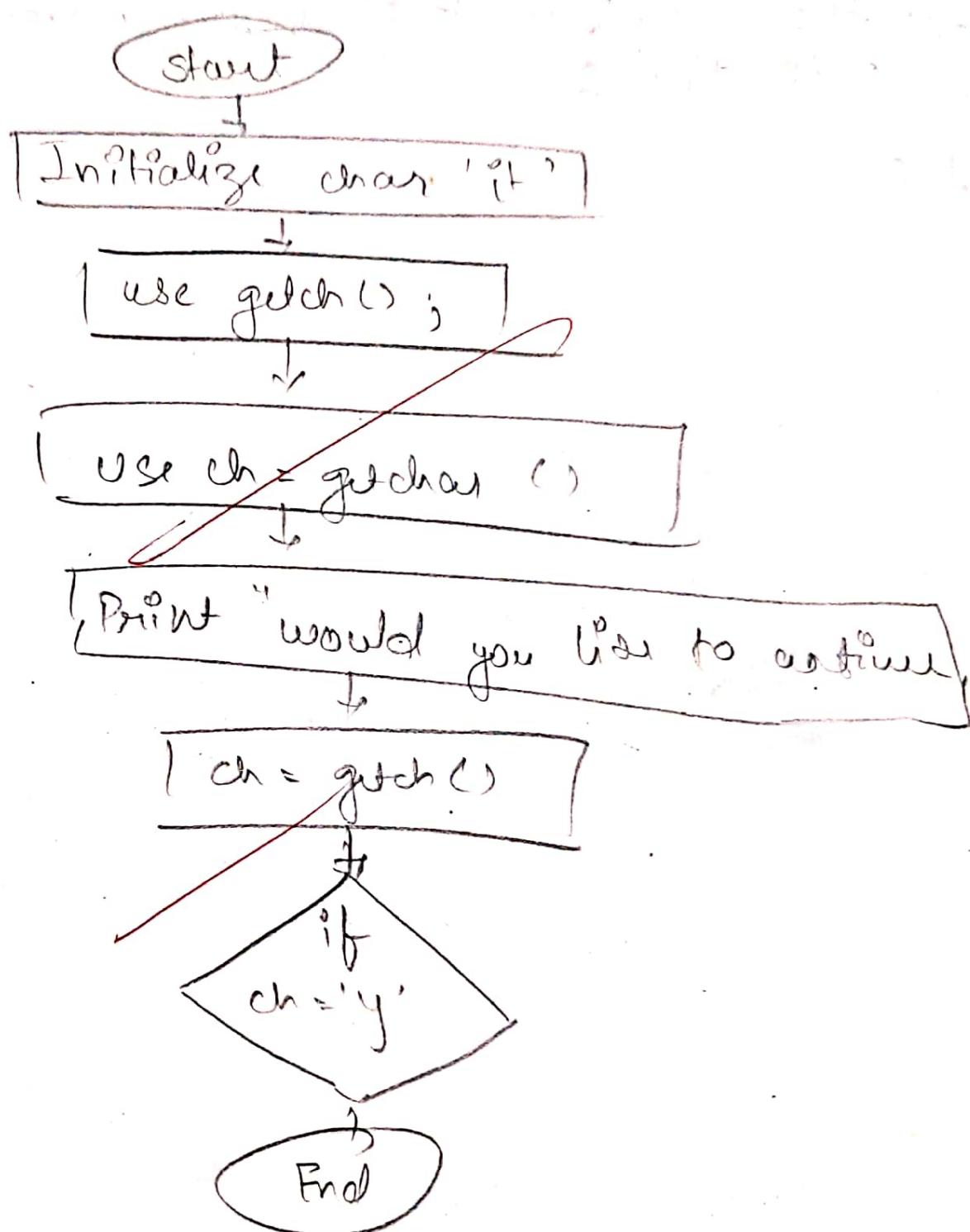
046

write a program which shows the use of get() function.

- step 1: Start the turbo C application.
- step 2: Initialise the character variable 'ch'
- step 3: Use the getch() method to accept the character
- step 4: Use the getchchar() method to show an option 'y/n' in 'ch'
- step 5: while 'ch == 'y', keep accepting value for 'ch'.
- step 6: Use the getch() method to store 'y' or 'n' in 'ch'.
- step 7: Terminate.

Conclusion:

Q40



580

```
#include <sfdio.h>
#include <conio.h>
void main()
{
    char ch;
    clrscr();
    printf("Press any key to continue");
    getch();
    ch = getch();
    printf("Would you like to continue\ny/n");
    ch = getch();
    while (ch == 'y')
    {
        printf("Would you like to continue");
        ch = getch();
    }
}
```

output:

048

press any key to continue

Enter any character: A

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

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

049

III WAP to show the use of put() function.

Step 1: Start the turbo C application.

Step 2: Initialize a character 'ch' to 'n'

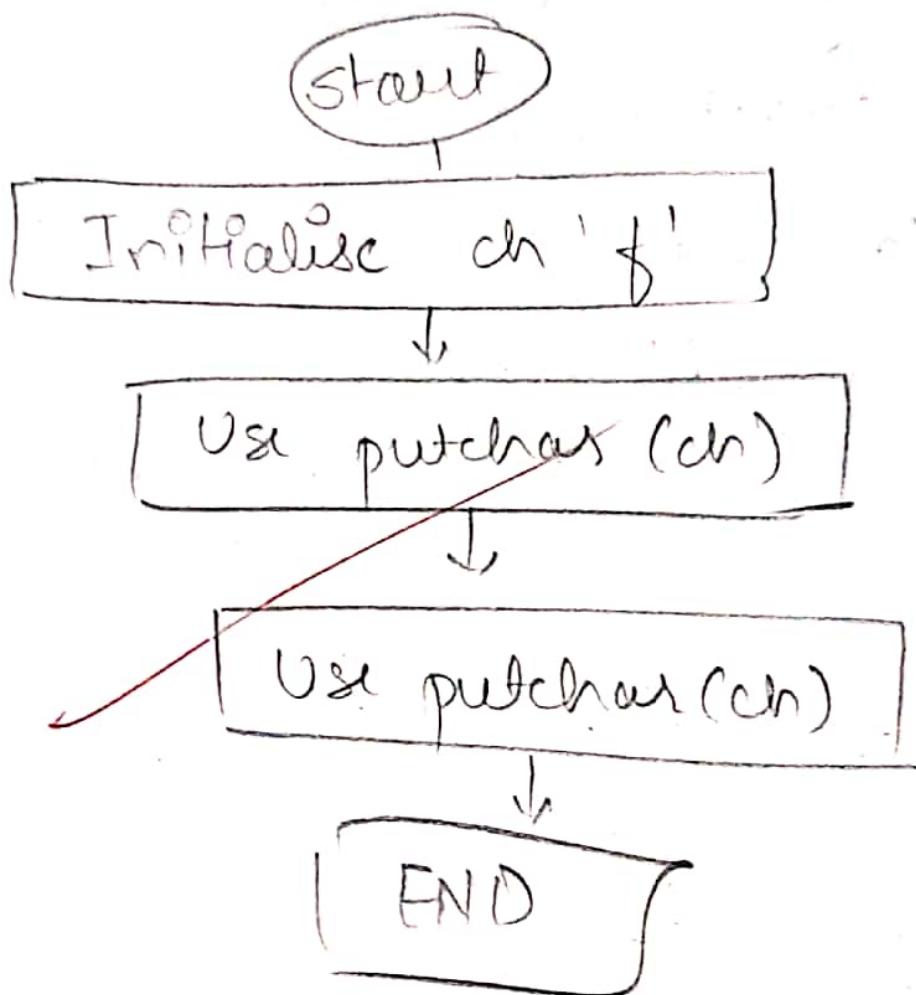
Step 3: Use the getch() and putchar() function with 'ch' as the argument.

Step 4: Terminate the program.

Conclusion: The ~~use~~ of factorial use of getch and put ~~c~~ () function has been done successfully.

Thmde

END



Q40

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

output:

A

A

### Practical No. 7

Aim : Write a C program to swap two numbers using pointers.

Algo :

Step 1 : Start turbo C application

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

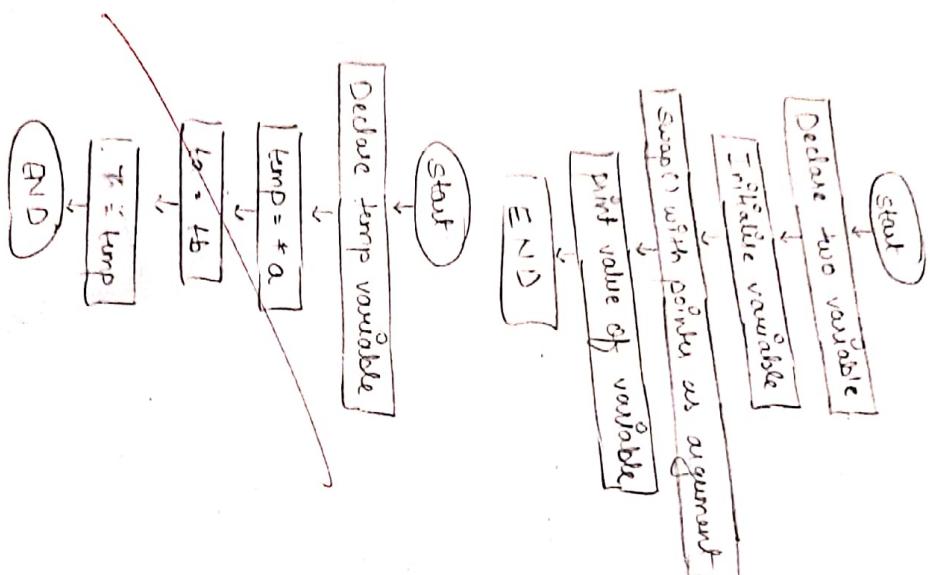
Step 3 : Declare two variable and accept the value from the user print the corresponding

value using printf()

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

Step 5 : Print the respective value of the variable.

Step 6 : Use the basis of swapping algorithm in the definition but instead of normal variable we



# Source code:

```
#include <stdio.h>
#include <conio.h>
void swap (int * a , int * b)
```

```
{  
    int k;  
    k = *a;  
    *a = *b;  
    *b = k;  
}  
int main ()  
{  
    int n1 , n2;  
    clrscr ();  
    printf ("Enter the number \n");  
    scanf ("%d" , &n1);  
    printf ("Enter the number \n");  
    scanf ("%d" , &n2);  
    printf ("Before swapping n1:%d , n2:%d  
           swap (&n1 , &n2);  
    printf ("After swapping n1:%d , n2:%d  
           getch ();  
}
```

Output:

Enter the number  
77  
88

Before swapping n1:77 , n2:88  
After swapping n1:88 , n2:77



Source 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++)
        {
            if (j = 0 : j < 10 ; j++)
            if (*a > *a + 1)
                if (temp = *a + 1,
                    *d + 1 = *a ;
                    *a = temp;
                }
        }
        printf ("%d is the sorted array", a);
        getch();
    }

```

Insert element into the array.

1 6  
7 8  
2 9  
10

11 12

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

Is the sorted array.

Algorithm:

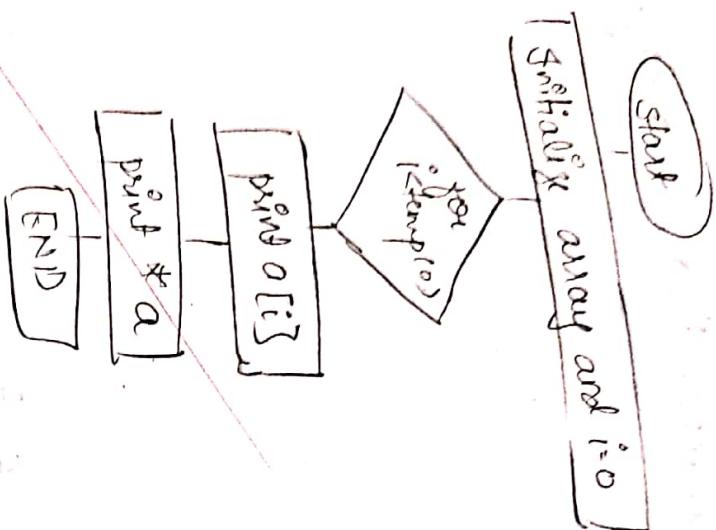
Step 1: Start the turbo C.

Step 2: Initialize an integer array and a variable i.

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

Step 4: Print the data of the array and the points to print at the memory location.

Step 5: Terminate the program.



code :

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

```
{ int a[10];
clrscr();
for ("Enter the element : ");
scanf ("%d", &a[i]);
for (i=0; i<10, i++);
}
printf ("%d", a[i]);
for (i=0; i<10, i++);
{
    printf ("Data : %d", *(a+i));
    printf ("Address : %d", &(a+i));
}
```

Data : 1  
 Address : 65506  
 Data : 2  
 Address : 65506  
 Data : 3  
 Address : 10  
 Data : 4  
 Address : 30  
 Data : 5  
 Address : 50  
 Data : 6  
 Address : 70  
 Data : 7  
 Address : 90

*Khushali:*

Conclusion : Thus pointers have been used for various operations.

Data : 6  
 Address : 70  
 Data : 7  
 Address : 90

### 8. Practical No. 8

Aim: Various operation on strings.

- Structure Student:

Algo:

Step 1: Open the turbo C application.

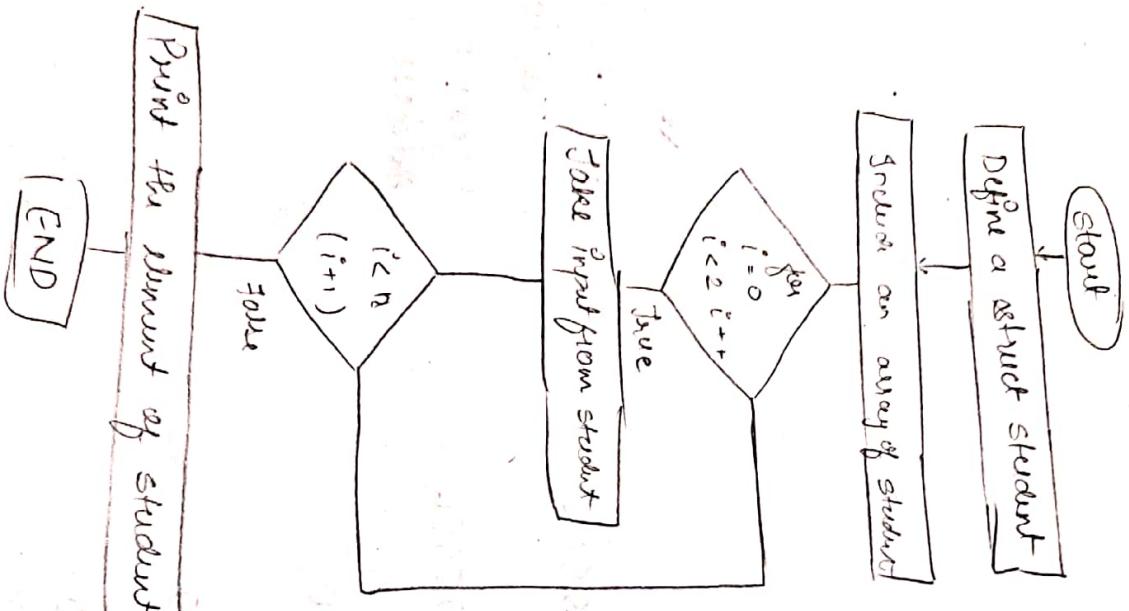
Step 2: Define a struct "student" with id, CGPA and name variables.

Step 3: In the main () function declare an array of student.

Step 4: Use the scanf () function to scan the various elements of the student struct.

Step 5: Print all the elements of the student structure using another for loop.

Step 6: Terminate the program.



Output:  
Enter the detail of student!  
1881

058

50.

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

struct student

{  
int id, CGPA;  
char name[10];

};  
void main ()

struct student s[2];  
int i;  
clrscr();

for (i=0 ; i<2 ; i++)  
{  
printf("Enter detail of Student %d , %d),  
scanf("%d", &s[i].id);  
scanf("%s", &s[i].name);  
scanf("%f", &s[i].CGPA);  
}

}  
for (i=0; i<2; i++)

Name : Smit.

printf ("Enter the Student Id", i+1);

printf ("In ID : %d\nCGPA : %d

Name : %s %c %d,  
s[i].CGPA , &s[i].name);

};  
getch();

Detail of student 1  
ID : 1881

CGPA : 8

Name : Dhruven.

Detail of student 2  
ID : 1862

CGPA : 9

(iii) Use of Struct with function.

Algo:

step 1: Open the Turbo C application.

step 2: Define a struct and take the input &

step 3: Now define a function to calculate total marks obtained out of 600 using CGPA.

step 4: Print this along with the student data.

step 5: Summarize the program.

Start

{Def a function to calc marks}

{Initialize a structur student}

{use the CGPA for  
stud student}

{Print the student detail and  
value of function}

END

(iii) Union

Algo:

Step 1: Start the Turbo C application.

Step 2: Use the union keyword to declare the union of different datatypes.

Step 3: In the main body of the function use the `scanf` to take the input.

The code of the union  
is given below

Use for loop for  
taking input

Steps

Step 4: Now print all the data of union.

Step 5: Terminate the program.

(END)

Output:

Enter the detail:

1881

Dhruven

B

9973587776

76.70%

The details of the student are:

Rollno: 1881

Name : Dhruven

Div : B

Contact no: 9973587776

Percentage: 76.70%

Code :

061

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int rollno;
    char name [10], div, contactno [10];
    float percentage;
}

void main ()
{
    Union Student S;
    printf ("Enter the details:");
    scanf ("%d,%s,%c,%f", &rollno,
           name, &div, &contactno, &percentage);
    printf ("In the details of the student are:");
    printf (" Roll no %d, %s", rollno, name);
    printf (" In Name %s, %c, %d", name, div, contactno);
    printf (" In Div %c, %d", div, contactno);
    printf (" In Contact no %d, %f", contactno, percentage);
    printf (" In Percentage %f, %f", percentage);
    getch ();
}
```

Practical No. 9  
Copy one string into another string.

Aim: To copy one string into another string.

Algo:

Step 1: Start the Turbo C application.

Step 2: Input string from user and store it in a variable.

Step 3: Declare another variable to store copy of first string.

Step 4: Run a loop from 0 to end of string  
loop structures should be like

```
for (i=0; text1[i] = '\0', i++)
```

Step 5: Inside the loop for each character  
in text1 copy to text2  
say text2[i] = text1[i].

Step 6: Finally after loop make sure the copied string ends with null character.

start

Input string from user

copy text1 and text2  
characters in characters

for  
i=0, text1[i] = '\0', i++

Inside the  
loop copy  
text1 into  
text2  
text2[i] = text1[i]

make sure that string  
is null terminated

Output:

Enter any string : Jhunee am 7 days in a week  
 first string : Jhunee am 7 days in a week  
 first string copy : Jhunee am 7 days in a week

Total characters copied : 26

Code :

063

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char text1[Max_size];
    char text2[Max_size];
    int i;
    printf("Enter any string:");
    scanf("%c", &text1);
    for (i = 0, text1[i] = '\0'; i++)
    {
        text2[i] = text1[i];
    }
    text2[i] = '\0';
    printf("First string = %s\n", text1);
    printf("First string copy = %s\n", text2);
    printf("Total characters copied = %d\n", i);
    return 0;
}
```