

ss C Programming Practical:

Aim:- Write a program to understand basic data type & input / output

Requirements : Turbo - C

Steps

1. This C - program creates a file and stores information
2. We frequently use files for storing information which can be processed by our program.
3. In order to store information permanently and reuse it we need to use files and this program demonstrate file creation and writing data in that.
4. The same code is used to create a file and store information
5. 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 data-type work in C & also about how we can give input & display output.

Program:

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

Output:

```
Enter your roll no: 1870
Enter your percentage: 75
Enter your name: Abhay
Enter your mobile no: 7030234355
Your roll no : 1870
Your percentage : 75
Your name : Abhay
Your mobile no : 7030234355
```

Program code:-

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, c;
    clrscr();
    printf("print 2 nos.");
    scanf("%d,%d", &a, &b);
    c = a + b;
    printf("The sum of nos is %d\n", c);
    c = a - b;
    printf("The sub of nos is %d\n", c);
    c = a * b;
    printf("The mult of nos is %d\n", c);
    c = a / b;
    printf("The div of nos is %d\n", c);
    c = a % b;
    printf("The modulo of nos is %d\n", c);
    getch();
}

```

Output:- Print two nos : 20,10

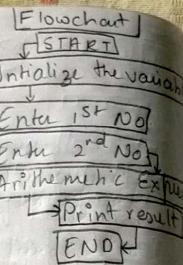
The sum of nos is: 30

The diff of nos is: 10

The mult of nos is: 200

The div of nos is: 2

The modulo of nos is: 0



Practical 2:

29

Aim:- Write a program on operator & expression

Requirement : Turbo C

Operators are foundation of any programming language. Thus the functionality of C-language is incomplete without the use of operator. Operator allows us to perform different kinds of operations on operands.

Arithmetic operators:-

Addition: The '+' operator adds 2 operands ($a+b$)

Subtraction: The '-' operator subtracts 2 operands ($a-b$)

Multiplication: The '*' operator multiplies 2 operands ($a*b$)

Division: The '/' operator divides 2 operands (a/b)

Modulus: The '%' operator returns the remainder when first operand is divided by second ($a \% b$)

Algorithm :-

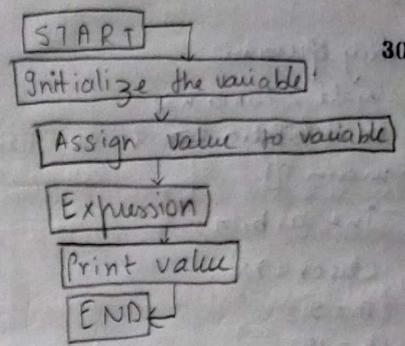
Step 1 - Declare a variable name for first and second no. of int.

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

Step 3 - Now to add the 2 nos. given by user, use the expression num1 + num2.

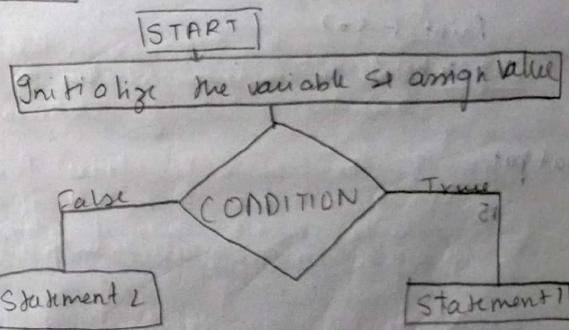
Step 4 - Now use printf function to display output.

Flowchart 2 :-



30

Flowchart 3:-



b) code:

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

Output:

15

W.A.R. in C to explain Ternary operator.

Algorithm:-

Step 1: Declare variables a, b & x as integer.

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 '?' to find.

Step 4: Use printf function to display output.

Conclusion: These programs helps in having better understanding about Operator & expression.

Practical no. 3

Aim:- WAP in C on decision statements
(if, if...else, nested if)

Theory:-

Algorithm:-

a) Step1: Declare a variable as integer and assign its value i.e. zero.

Step2:- Now to compare whether 20 is greater than 15 use if statement.

Step3:- If the condn is true, printf "20 is less than 15" else if condn is false skip the if statement & print "20 is not in if".

B) WAP in C to explain if else statement

Step1:- Declare a variable as integer and assign its value i.e. 20

Step2:- Now to compare the given value is greater or not use if condition.

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");
    }
}
```

Output: g am not in it

Output: g am not in it

b) Code:

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

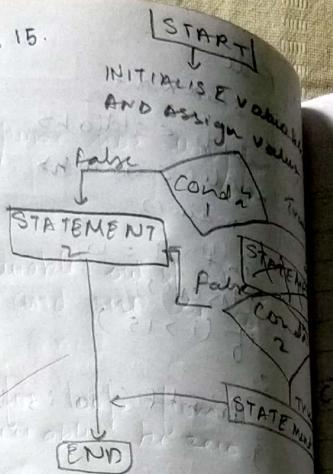
Output: 20 is greater than 15.

C] CODE:

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

Output:

20 is greater than 10 & 12.



33

Step 3: If condition is true print 20 is less than 15 or if cond1 is true false then print 20 is greater than 15.

WAP in C to explain nested if

Algorithm:

Step 1: Declare a variable i as integer & assign value zero.

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

Step 3: If cond1 is True goto second condition then 20 is greater than 15 & if one of condition are not True then skip the part & print 20 is greater than 5 & 12.

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

Notes

Practical 6:-

P- Aim:- To display the prime nos using for loop

Algorithm:-

Step 1:- Initialise the variable out of which 2 are loop variable and one as count variable.

Step 2:- Initialise a for loop from i to n so the count variable be zero.

Step 3:- Nest another loop within the loop in Step 2 that goes 2 to the first variable i.e. 2, 4, 6, 8, ...

Step 4:- Use the if condition statement to check whether (first loop variable) & (2nd loop variable) is true increment by count by 1.

Step 5:- Come out of 2nd loop & check whether the count variable is 0, if true print No.

Step 6:- Terminate the program.

Conclusion:- Thus the prime nos were printed using for loops

Code:-

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

```
int n, i, a;
```

```
clrscr();
```

```
printf("The prime no. 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 nos are:- 2 3 5 7 11 13 17

No.

```

> Code
#include <conio.h>
#include <stdio.h>
Void main()
{
    int n1=0, n2=1, n3, i, num;
    clrscr();
    printf("Elements total");
    scanf("%d", &num);
    printf("%d %d", &n1, &n2);
    for (i=2; i<num; i++)
    {
        n3=n1+n2;
        printf("%d", n3);
        n1=n2;
        n2=n3;
    }
    getch();
}

```

Output:-

Element total: 10
 0 1 1 2 3 5 8 13 21 34 55

Aim:- WAP in C to display fibonacci series.

Step 1:- Start turbo c

Step 2:- Declare n1, n2, n3, i, number variables
 Step 3:- Initialise the variable n1=0, n2=1 and number=0.

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

Step 5:- Print first 2 terms of series as n1=0 & n2=1.

Step 6:- Use the for loop as per foll step:

n3 = n1 + n2; n1 = n2
 n2 = n3

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

Step 7:- Print the value of nos.

Step 8:- End

Conclusion:- Thus we have successfully executed fibonacci series on turbo C.

28

C] Aim :- WAP in C on foll expression

```

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

Algorithm:-

Step 1 - Start Turbo C

Step 2 - Declare the variable i, j, rows;

Step 3 - Display the no. of rows

Step 4 - Enter the for loop as i=1; i<=rows;

Step 5 - Create nested for loop as j=(i,j<i);

Step 6 - Display the nos. as per requirement
in sequence from i=1.

Step 7 - Inc. the no. to i++ form

Step 8 - Display the output

Conclusion:- Thus we have successfully
written given expression in Turbo C using nested for loop

Ans

code

```

#include <conio.h>
#include <stdio.h>
void main()
{
    int i, j, rows, num=1;
    clrscr();
    printf("Enter no. of rows \n");
    scanf("%d", &rows);
    for(i=1; i<=rows; i++)
    {
        for(j=1; j<=i; j++)
        {
            printf("%d ", num);
            num++;
        }
        printf("\n");
    }
    getch();
}
    
```

Output:

```

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

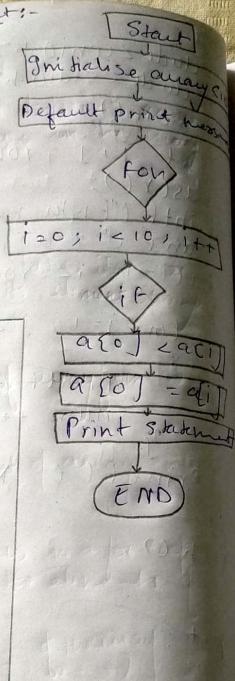
No.

```

Coding:-
//C program to find largest
//array element
#include <cs51.h>
#include <conio.h>
void main()
{
    int i, n;
    clrscr();
    float arr[100];
    printf("Enter the no. of elements\n");
    printf("from 1 to 100");
    for(i=0; i<n; i++)
    {
        printf("Enter no %d", i+1);
        scanf("%f", &arr[i]);
    }
    for(i=1; i<n; i++)
    {
        if(arr[0] < arr[i])
            arr[0] = arr[i];
    }
    printf("Largest Element = %.2f", arr[0]);
    getch();
}

```

y
 Output
 Enter the element of the array
 12, 23, 2, 2, 12, 55, 3, 1, 2, 100
 The largest no. : 100

Flowchart:-

37

Practical 5

Aim:- WAP in C to find largest no. using array.

Algorithm

Step 1:- Start turbo C application

Step 2:- Declare the application 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, use the if conditional statement check
 if a[0] < a[i] if true put
 a[0] = a[i]

Step 5:- Run the for loop. Terminate the program

Conclusion:- Hence, This program was successfully submitted.

CO 88

Ques Aim: Write a C-program to print a no
of odd & even no. in the array

Algorithm:

Step 1: Create an array, takes its size from user & define its element using loop.

Step 2: Display the size of array from the user.

for
{
 p Step 3:- Display the element of array entered
 by the user
 sc
}

for
{
 i Step 4:- Take the initialisation for loop
 using which all the elements of the
 array are
 sc
}

Step 5:- Display the even no. of the array
loop (if array[i] % 2 == 0)
print "Even no."

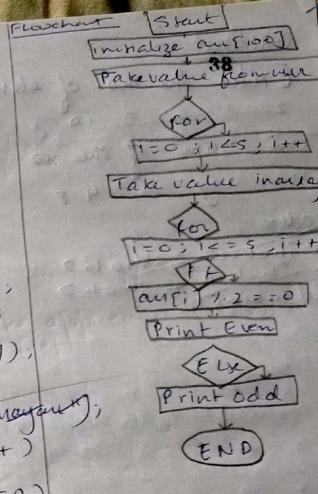
Step 6:- Else print "Odd no"

Step 7:- Terminate the turbo C application

Step 8:- Hence this program successfully executed.

Code -

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int arr[100], i, num;
    clrscr();
    printf("Enter the size of array (n)");
    scanf("%d", &num);
    printf("Enter the elements");
    for(i=0; i<num; i++)
        scanf("%d", &arr[i]);
    printf("Input has been entered");
    for(i=0; i<num; i++)
    {
        if(arr[i] % 2 == 0)
            printf("Even No %d", arr[i]);
        else
            printf("Odd No %d", arr[i]);
    }
}
```



Output:- Enter the no. of elements: 5

1, 4, 6, 8, 9
Even no:-
4, 6
Odd no:-
1, 9

Output:-

Enter the no. of element : 10

2 3 6 5 10 1 2 1 7 11

The sum of the no. is 48

The avg is 4.8

Code:-

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

```
void main()
```

Aim:- WAP to find sum & average
and average.

algorithm:-

- Step 1:- Start turbo c application.
- Step 2:- Declare the int variable n, i, sum Initialize {num[100]}, sum = 0.0, avg.
- Step 3:- avg using for loop at i=0, i<n, i++
give print message and increment i by 1.
- Step 4:- Declare sum variable and store it by adding num[i].
- Step 5:- Average is sum divided by no.
- Step 6:- Give print statement for average and sum.
- Step 7:- Terminate the program.

Conclusion :-

Thus we have executed the program successfully.

Nimadu

source code :-

```
#include <iostream.h>
#include <conio.h>
void main ()
{
    int n, i;
    float num[100], sum=0.0, avg;
    clrscr();
    cout << "Enter the no. of elements ";
    cin >> n;
    for (i=0; i<n; i++)
    {
        cout << "Enter no." << i+1 << endl;
        cin >> num[i];
        sum = sum + num[i];
    }
    avg = sum/n;
    cout << "Average = " << avg;
    cout << "sum = " << sum;
    getch();
}
```

Flowchart

Start

Define a function called factorial with argument
n as integer

$n \geq 1$

False

Return 1

True

$n = \text{fact}(n-1)$

Return

Now we main() & declare variables
a, n as integers

Print no. to find its factorial

Call factorial function

Display output

END

Output: Enter a no. to find its factorial.

Factorial of 5 is 120

41

Practical @ No. 6.

APM = Program on function.

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

Algorithm:

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

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

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

Step 4: Now enter number to find its factorial.

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

Step 6: Terminate the program.

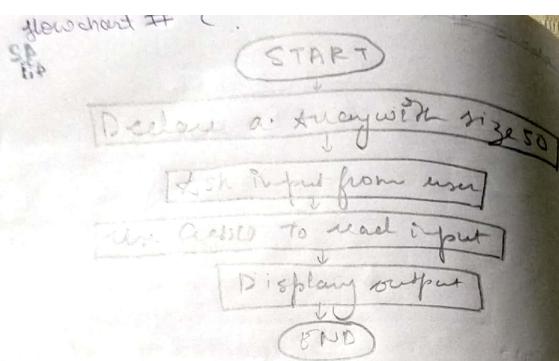
Source code:

```
#include <stdio.h>
#include <conio.h>
int fact (int n)
{
    if (n >= 1)
        return n * factorial(n-1);
    else
        return 1;
```

3
void main ()

```
{     int a,n;
      printf("Enter a no to find its factorial");
      scanf("%d", &n);
      a = factorial(n);
      printf("The factorial %d is %d", a);
      getch();
```

Conclusion: Successfully executed the program to find factorial using recursive function.



output # 2

Enter your name : Veeru
 Your name : Veeru

43

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

Algorithm:

Step 1: Declare a variable or name as array with size so with character datatype.

Step 2: Now ask your name by user.

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

Step 4: Print your name on screen.

Step 5: Test & run the program.

Program:

```

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

Conclusion: successfully executed the program.

Q.3) Write a C program to show the use of function.

Flowchart

Algorithm:

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

Step 2: Now ask user to enter your name.

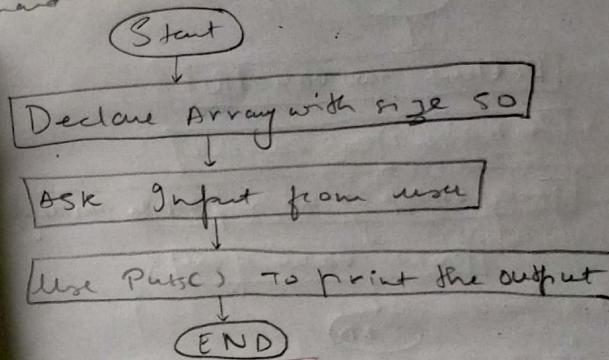
Step 3: Now use puts function to display the output.

Step 4: Terminate the program.

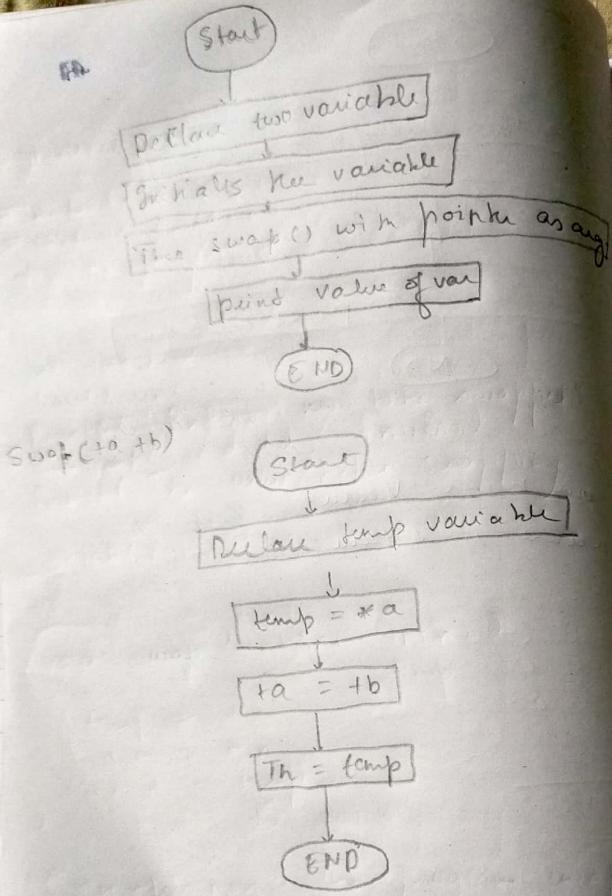
Code:

```
#include <string.h>
#include <stdio.h>
#include <conio.h>
void main()
{
    char name[50];
    printf("Enter your name.");
    scanf("%s", &name);
    printf("Your name is %s", name);
    puts(name);
    getch();
}
```

~~Conclusion:-~~ Successfully executed the program



Output:-
Enter your name Veeru
Your name is : Veeru



Practical - 7

Aim:- Write a program to find the swapping of 2 nos.

Algorithm:-

Step 1:- Start the Turbo C application.

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

Step 3:- Declare 2 variable & accept their values from the user. Print the value using printf()

Step 4:- Pass the address of variable as argument for the function

Step 5:- Print the required value of the variable

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

No. C
4
Program :-

```
#include <conio.h>
#include <stdio.h>
void swap (int *m, int *n)
void main ()
{
    int m, n;
    clrscr();
    printf("Enter 2 nos. to be swapped");
    scanf("%d %d", &m, &n);
    printf("The value before swapping %d %d", m, n);
    getch();
}

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

Output :-

Enter 2 nos. to be swapped : 12, 24
 The nos. before swapping are : 12, 24
 The nos. after swapping are : 24, 12

Sorting of arrays using pointers
Algorithm:-

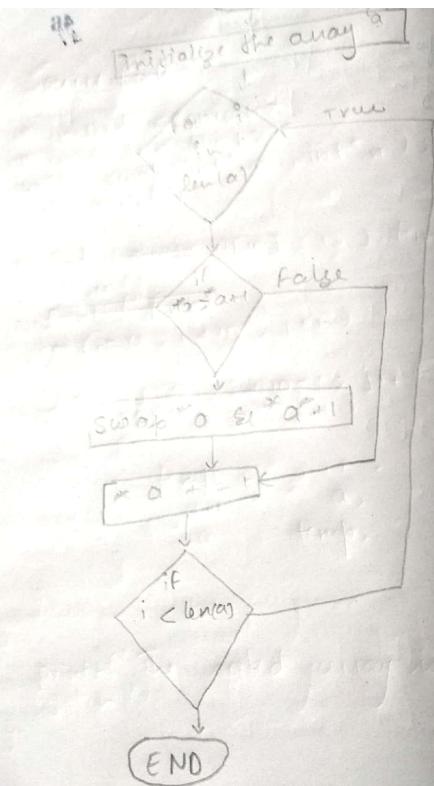
Step 1:- Initialise an integer array, $a[]$
Set temp variable

Step 2:- Run a nested for loop of $i=0$ to $\text{len}(a)$
and of $j=0$ to i $\text{len}(a)-i$.

Step 3:- If $a[j] > a[j+1]$, swap the values using
basic swapping method

Step 4:- Print the swapped array

Step 5:- Terminate the program.



Source Code:-

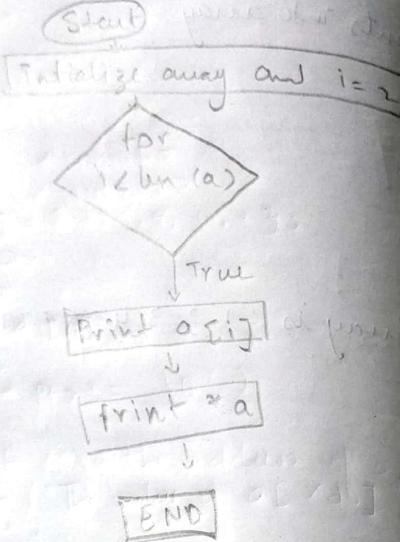
```
#include <conio.h>
#include <stdio.h>
void swap(int* m, int* n);
void main()
{
    int a[10], i, j, temp;
    clrscr();
    printf("Enter");
    for(i=0; i<10; i++)
    {
        for(j=0; j<10-i; j++)
        {
            if(*a > *a+1)
            {
                temp = *a+1;
                *a = *a+1;
                *a+1 = temp;
            }
        }
    }
    printf("The sorted array is : %d", a);
    getch();
}
```

Output :-
Input elements into array :

48

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

Flowchart:-



49

WAP to find one dimensional array using pointers.

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. Then use pointer to print the memory location.

Steps:- Terminate the program.

Code

```
# include <conio.h>
# include <stdio.h>
int a[5] = { 7, 5, 9, 4, 2 };
int i = 0;
int *ptr;
ptr = &a[0];
clrscr();
while (*ptr != '10')
{
    printf ("\n the address of a[%d] = %u", i, *ptr);
    printf (" The value of a[%d] = ", i, *ptr);
    ptr++;
    getch();
}
```

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

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

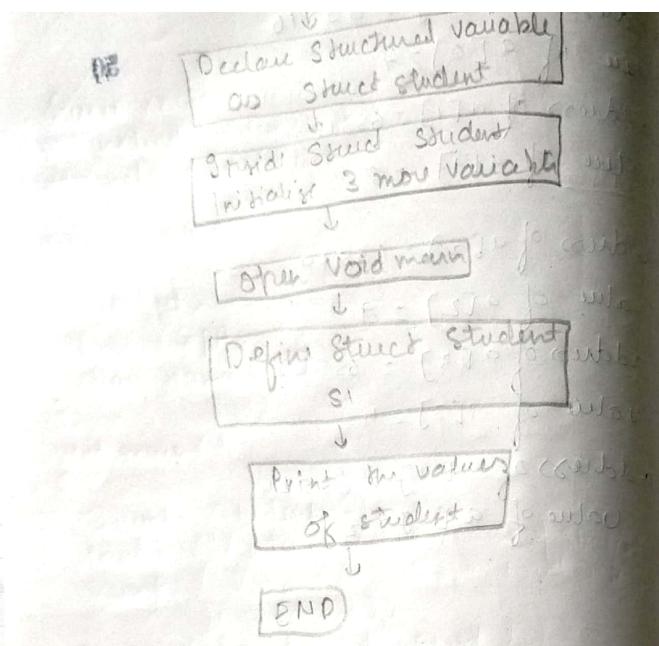
The value of a[2] = 9

The address of a[3] = 65522

The value of a[3] = 4

The address of a[4] = 65524

The value of a[4] = 2



Practical 8:

Aim:- Create a simple structure named as student that holds following variables id , CGPA , Name

Step 1:- Start Turbo C application

Step 2:- Declare the structure variable as struct-

Step 3:- Initialise the struct variable with 3 more variables 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 of the students such as id-CGPA , Name

Step 6:- Terminate the program.

Some code:-

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

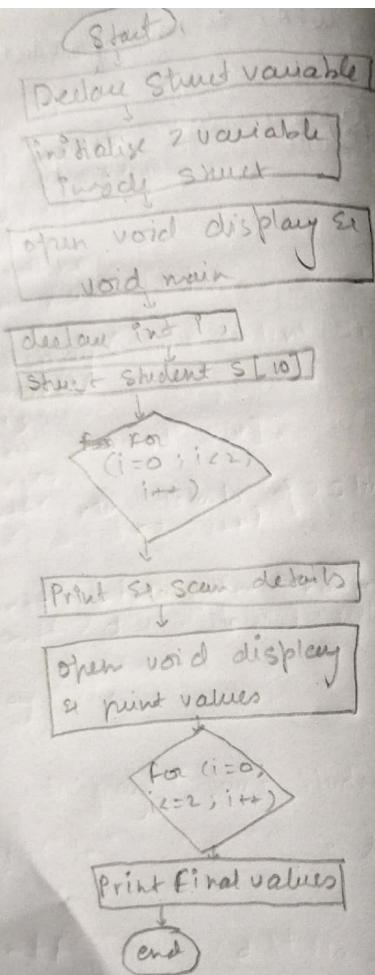
```
struct Student
{
    int id;
    float CGPA;
    char name[10];
};

void main()
{
    struct Student S1;
    printf("Enter id CGPA, name of student: ");
    scanf("%d,%f,%s", &S1.id, &S1.CGPA, &S1.name);
    printf("\n id = %d", S1.id);
    printf("\n CGPA = %.2f", S1.CGPA);
    printf("\n name = %s", S1.name);
    getch();
}
```

Input:-
 id = 1
 CGPA = 8.333
 name = Abhay

Flowchart:-

55



Aim:- WAP which will demonstrate use of struct
and function

53

Algorithm:-

Step 1:- Start Turbo C application.

Step 2:- Declare the structure variable as
struct

Step 3:- Initialise the struct student with
2 or more variables (int roll no 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 students upto 2 students if not
more than that.

Step 6:- Print the details of student.

Step 7:- Open void display again &
print the values using for cond % a
printf

Step 8:- Terminate the program.

83
 Code:-

```
#include <conio.h>
#include <stdio.h>
struct Student
{
    int roll_no;
    char name[10];
};

void display( struct Student s1[10] )
void main()
{
    int i;
    struct Student s[10];
    clrscr();
    printf("Enter details of 2 students");
    for (i=0; i<2; i++)
    {
        printf("nEnter roll no. & name");
        scanf("%d %s", &s[i].roll_no, s[i].name);
    }
    display(s);
}

void display( struct Student s1[10] )
{
    int i;
    printf("\n* * * * *\n");
    for (i=0; i<2; i++)
    {
        printf("Roll = %d Name = %s", s1[i].roll_no, s1[i].name);
    }
}
```

out put:
 details of 2 students
 n Enter Enter
 Enter Enter
 Enter Enter
 * * * * *
 Roll = 22 Name = o
 Roll = 33 Name = prakash

Enter any string : There are seven day in a week
first string : There are seven day in a week
First string copy : These are seven day in a week
Total characters copied : 26.

Practical 9:-

55

Aim:- WAP to copy one string to another

Algorithm:-

Step 1:- Input String from user & store it to some variable say text_1 .

Step 2:- Declare another variable to store copy of first string text_2 .

Step 3:- Run a loop from 0 to end of string. The loop structure should be like :-

```
for ( $i=0$  ;  $\text{text}_1[i] = '0'$  ;  $i++$ )
```

Step 4:- Inside the loop for each char in text_1 copy to text_2 . say $\text{text}_2[i] = \text{text}_1[i]$

Step 5:- Finally after loop make sure the copied string ends with Null character i.e. $\text{text}_2[i] = '0'$;

Code:-

```
# include <stdio.h>
# include <conio.h>
int main ()
{
    char text1 [max-size];
    char text2 [min-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 copy = %s\n", text2);
    printf ("Total character copied = %d\n", i);
    return 0;
}
```

56
Input string from user

copy text1 to text2
character by character

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

Start

int main

Defines a
string

use the
strcat function

Print the concatenated
Result

end

output: Rahul is 18 year old

57

Write a program which will demonstrate
the use of string library functions.

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

The strcat function returns a pointer
where the resulting concatenated string resides.

* Code

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

2

3

start

87 take string as input

start the string
in array

use for loop
to count no. of
char in array

```
for  
(i=0;  
string[i]!='\0'; i++)
```

store result in
variable

Print variable as
output

end

59

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

Algorithm:- i) Take a string as input and store
it in the array

ii) Using for loop count the no. of characters
in array and store the result in
variable

iii) Print variable as output:

Code:-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char str[50];
    int i, length=0;
    printf("Enter a string\n");
    gets(str);
    for (i=0; str[i] != '\0'; i++)
    {
        length++;
    }
}
```

printf("The length of str is %d\n", length);

printf("So the length is : %d = %d\n", length);

Output:-

Enter a string
is a cold night.

length of str is the no. of characters init
the length of: is a cold night = 18

60

Read from the text file & close it

60

student
initialize int
num & file *ptr
in int main

if
(fptr = fopen ("C:\Program files\\",
"r")) == NULL)

Program exits if
file pointer returns

Scan & print
value of file to
read it

close the file
to read it

end

Practical 10:

61

Aim:- Program for file open , filenad,
& file close

fopen () → opens a existing file or
create a new file

frread () → Reads a record from a file

fclose () → Closes a file.

Ques

Code :-

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

Output :

```
values are = 87
88
89
90
```

Start

52

Declare char c
in int main

Print the
statement for
entering character

while
((!= 's');

un c = fgetchar ()
function

Print the result

end

63

Aim :- WAP for fgetc(), fgetchar(), fgetch()

Algorithm

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
exists \n");

while (c != '\$');

{

c = fgetchar();

printf ("In Enter character is : ");

putchar (c);

printf ("\n");

y

return 0;

y

Output :- Enter some character Enter \$ to exit

A

Enter character is : A

B

Enter character is : B

\$

output :-
opening the file test.c in read mode
Reading the file test.c
Hi, how are you?
close the file test.c

65

getchar() → Use to read a character from file Read single character at a time
In a program we use getchar() function
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;
```

```
    while (1)
```

```
        c = getchar (fp);
```

```
        if (c == EOF)
```

```
            break;
```

```
        printf ("%c", c);
```

```
}
```

printf("closing file test.\n");
f close (fp);
return 0

start

Declare file *fp
and char c
in int main

66

use open file
function

if

fp == NULL

could not open
file

while !

(c = fgetc (fp)) !=

'c' (c = liof)

close file

end