Prac - 01

Aim : Write a C program to understand basic data types & I/O

Theory :

Q] Write a program to display student's name, rollno, mobile no & percentage.

Algorithm

Step 1 : Declare a variable name rollno as integer also declare name, mobile no as char & percentage as float.

Step 2 : Use printf function to print questions 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

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

Unsaadu

printf ("Student's mobile no : %.8lf\n", mobile_no);

printf ("Student's percentage : %f\n", percentage);

getch ();

}

Output:

Enter Student's name :

Esha

Enter Student's roll no :

1892

Enter Student's percentage :

81.999

Enter Student's mob no :

98200 21021

Student's name : Esha

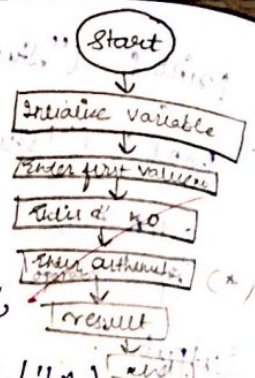
Student's roll no : 1892

Student's percentage : 81.999

Student's mob no : 98.200 21021

a). Code

```
// dynamic calculator
#include<stdio.h>
#include<conio.h>
void main()
{
    int num1, num2;
    float add, sub, mult, div;
    clrscr();
    printf("Enter first no :\n");
    scanf("%d", &num1);
    printf("Enter second no :\n");
    scanf("%d", &num2);
    add = num1+num2;
    sub = num1-num2;
    div = num1/num2;
    printf("Addn 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();
```



Prac: 02

Aim: 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 & second no as integer.

Step 2: Now use scanf funct^n to receive input from user.

Step 3: Now to add two no given by user use the expression num1 + num2

Step 4: Now to substract 2 nos given by use, use expression num1 - num2

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

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

Step 7: Now use printf function to display output.

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

Algo

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

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

Step 3: Now to compare "a & b" value is greater use ternary operator to find x.

Step 4: use printf function to display output

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

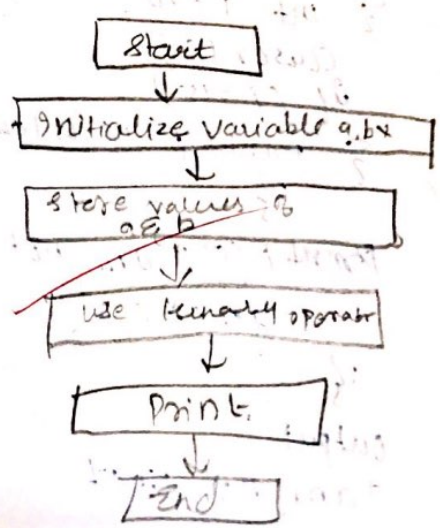
Viva

output

Enter first no: 3
Enter second no: 3
Addn of 3 & 3 is 0
Multiplication of 3 & 3 is 9.0000
Division of 3 and 3 is 1.000

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();
}
```



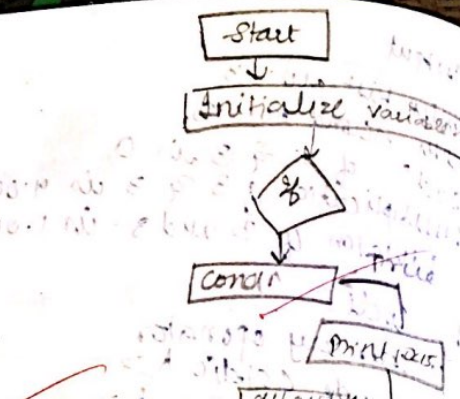
output

15

a] Code.

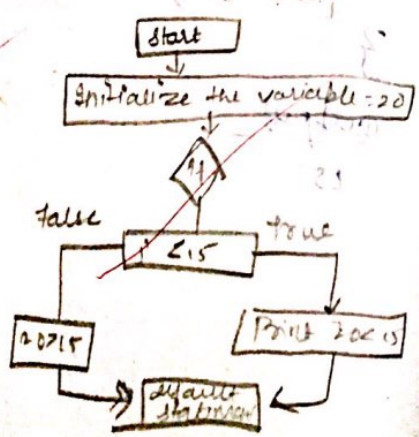
```
// if statement
#include<stdio.h>
#include<conio.h>
void main()
{ int i=10;
  clrscr();
  if (i>15)
  { printf("10 is less than 15 \n");
    }
printf("I am not in if \n");
    getch();
}
```

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");
  }
```



Prac: 03

Aim: write a prog in C on decision statement (if, else, nested if).

Theory:

a] write a prog in C to explain if statement

Algo

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

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

Step 3: If the condition is true, print- if condition is false skip the I am not in if.

b] write a prog 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 if it is greater or not use if else conditional statement.

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

c] write a prog in C to explain nested if statement.

Algo:
Step 1: Delare a variable as integer and assign it's 20

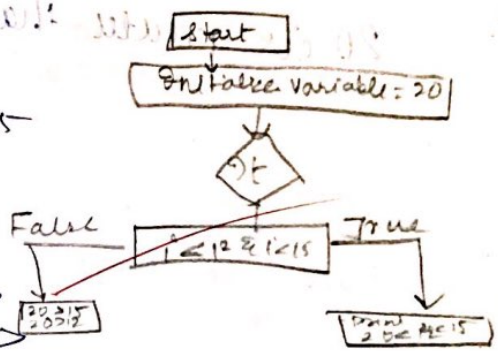
Step 2: Now use nestedif logic to compare if given no is greater or not.

Step 3: If first condition is true then go to second condition if 2nd cond is also true then print first 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<12)
{ printf("20 is less than 15 & 12 \n");
}
}
else:
{ printf ("20 is greater than 15 & 12 \n");
getch();
}



Output:

20 is greater than 10 & 12

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

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

Nirmadi

Practical: 04

Aim: To display prime numbers using for loop

Algorithm

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

Step 2: Initiate a for loop from 2 to 20. Let the count variable to zero.

Step 3: Next another loop within the loop (first) in step 2 that goes from 2 to that loop goes from 2. The first loop variables /2.

Step 4: Use the if conditional statement to check whether (1st loop variable == 0) if true, increment count variable by 1.

Step 5: Come out of the record loop and check whether the count variable is 0. If true, print the number (1st loop variable)

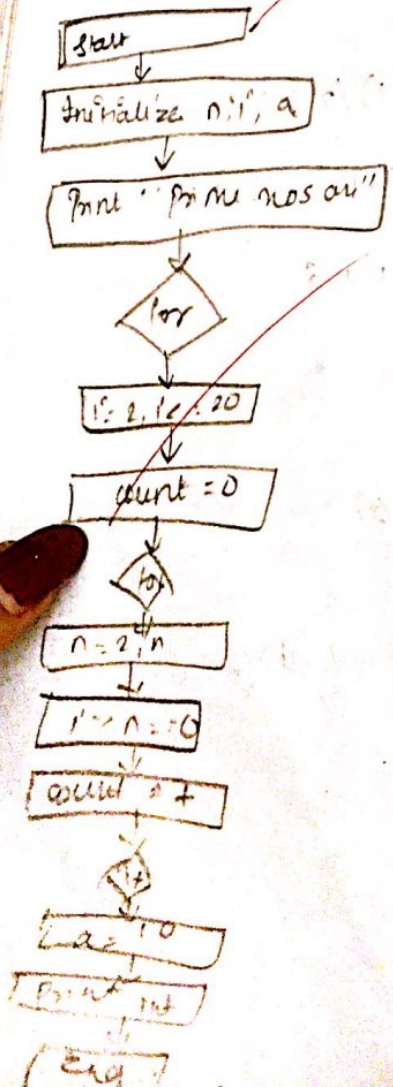
Step 6: Terminate the program

Conclusion: Prime nos were displayed using for loop

```
#include <conio.h>
#include <stdio.h>
void main()
{
  int n, i, a;
  clrscr();
  printf("The prime nos 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 nos are 2
3
5
11
13
17
19



→ WAP on fibonacci series

ALGORITHM

Step 1: Start the turbo C

Step 2: Declare the variable n_1, n_2, n_3, i and ina.

Step 3: Initialize the variable $n_1 = 0$, $n_2 = 1$ and $n_0 = 0$

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

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

Step: $n_3 = n_1 + n_2$

$n_1 = n_2$

$n_2 = n_3$

increase the value of i element of i and add element each time by 1

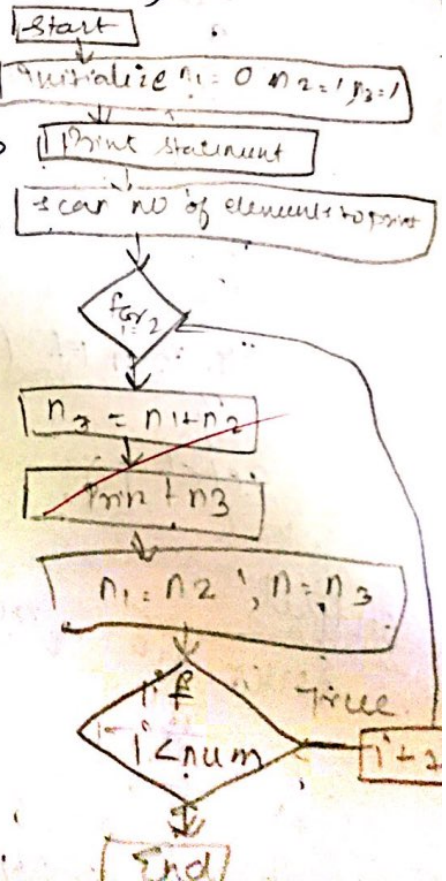
Step 7: Print the value of number

Step 8: End the program

Conclusion: Thus we have succesfully executed the program in turbo C

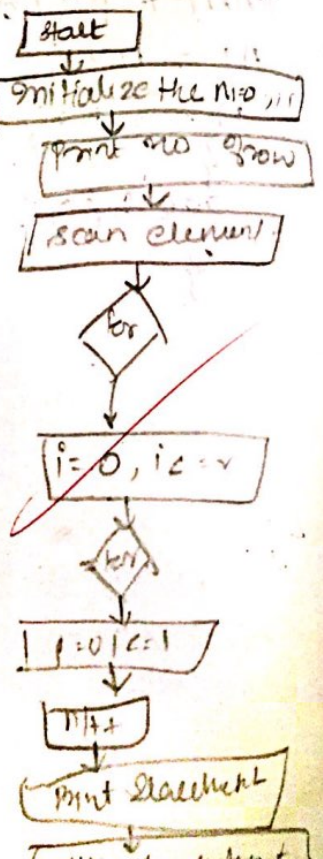
Code

```
#include <stdio.h>
#include <conio.h>
void main ()
{
int n1=0, n2=1, n3,i, number;
clrscr();
printf("Enter no of element :=\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();
}
```



#CODE

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n1=0, r, i, j;
    clrscr();
    printf("Enter the numbers 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 no of rows

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

41

Aim: write a C program on following expression

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

Algorithm

Step 1: start the turbo C program
Step 2: Declare the variables rows, i, j, number=1,
Step 3: Display the no of rows
Step 4: Enter the for loop and i=1, i<=rows
i++
Step 5: Create nested for loop j=1; j<=i;
++j
Step 6: Display the no as per user entry.
(i=)
Step 7: Implement no increment to i++ from
Step 8: Display the space.

Conclusion: Thus, we have successfully write given expression on turbo.C using nested for loop.

Anandi

Practical : 05

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

Algorithm

Step 1 : Start turbo C application

Step 2 : Declare the variable i and integer array a[].

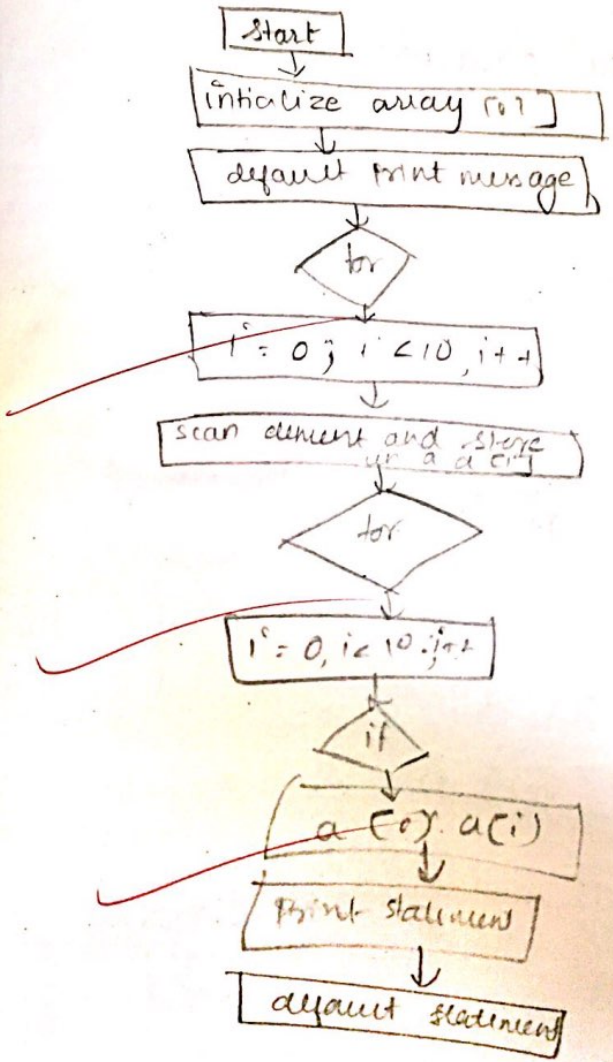
Step 3 : Enter the for loop at i=0 i<10 and scan the ele a(i) with i<10. Exit the for loop.

Step 4 : Enter the for loop at i = 0 i<10. use of condition statement to find if a[0] a[i] if true put a[i].

Step 5 : Run the above for loop for i<0, exit for loop.

Step 6 : Terminate the program.

FLOW CHART



Output

Enter the elements

1
25
3
17
77
3
100
100

The largest number is 100.

Source code:

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a[10], i;
clrscr();
printf("Enter the elements of the %d \n");
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 nos is:%d", a
getch();
}
```

Aim: C program to find even and odd nos in array.

Algorithm:

Step 1: Start Turbo C application Print the message for entering element. Store it in num variable.

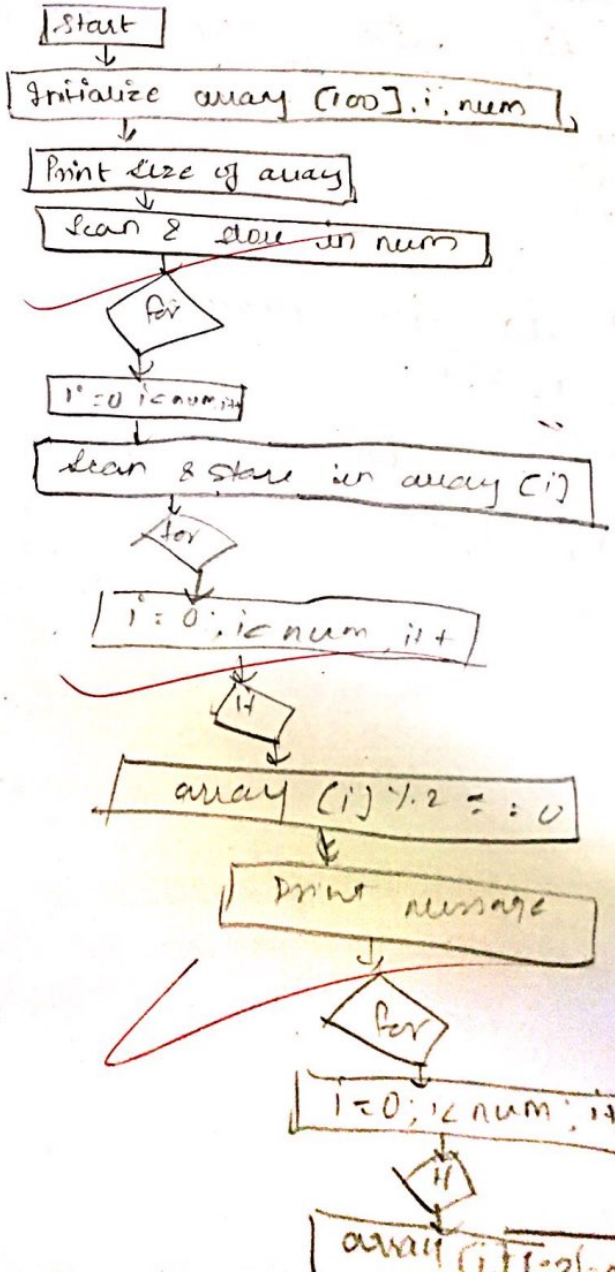
Step 2: Turn the for loop at i = 0; i < num and increment store the element in array[i].

Step 3: Give the print message for even nos use for loop at i = 0; i < num; i++ and use if condition to check the array element is odd.

Step 4: Give the print message, Exit for loop.

Step 5: Terminate the program

Flowchart



Output:

Enter the size of the array: 5
Enter the element of array: -

24
25
321
21
44

Even no in array are
24, 44

odd no in array are
25, 321, 21.

Source code

```
#include <stdio.h>
#include <conio.h>
{
int array [100], i, num;
printf ("Enter the size of the array\n");
scanf ("%d", &num);
Printf ("Enter the element of array\n");
for (i=0; i<num; i++)
{
scanf ("%d", &array[i]);
}
printf("Even no in the array are:");
for (i=0; i<num; i++)
{
if (array [i]%2==0)
{
Printf ("\n odd no. in the
array are :");
for (i=0; i<num; i++)
{
if (array (i) %2!=0)
{
printf("%d\t",array[
}
getch();
}
```

Aim : C program average and sum using arrays.

Algorithm

step 1 : Start turbo C application

step 2 : Declare the int variable n, i initialize num[100] sum = 0.0, avg.

step 3 : Using for loop ati=0. i<n, i++ Give print message and increment i by 1.

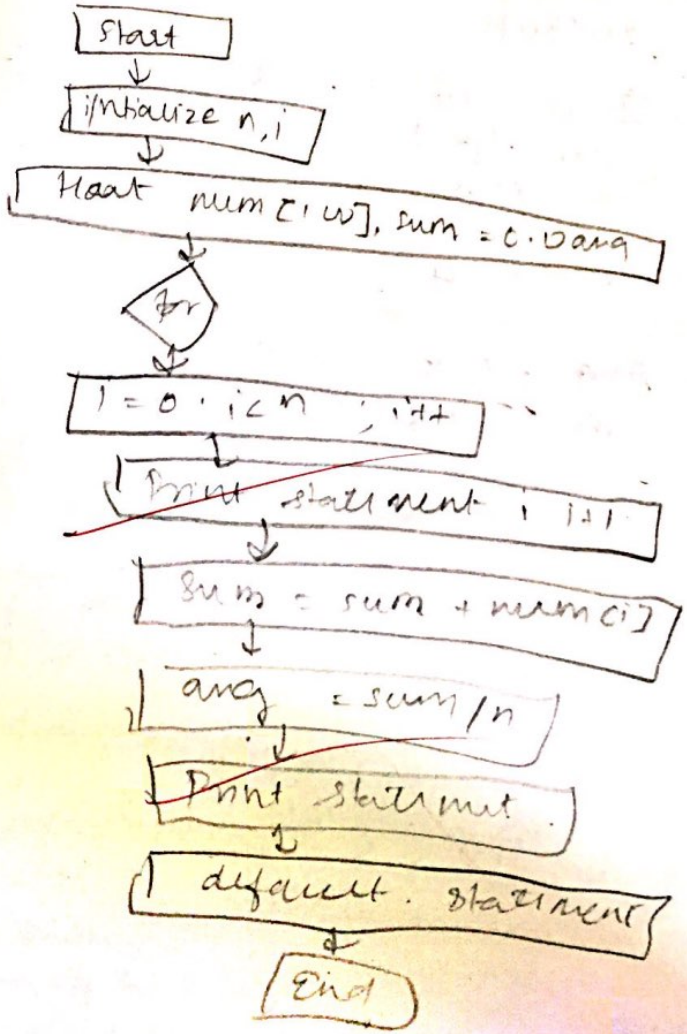
step 4 : Declare sum variable and store it by adding num(i)

step 5 : Avg is sum divided by n

step 6 : Give print statment for avg & sum.

step 7 : Terminate the program

Conclusion : Thus we have executed the program sucessfully



Output

Enter the no of elements :- 4
Enter no 1
Enter no 2
Enter no 3
Enter no 4

Avg = 4.5 0000
sum = 18.00.

Source code:

```
#include <stdio.h>
#include <conio.h>
Void main()
{
int n, i;
float num[100], sum = 0.0 avg;
clrscr();
Printf("Enter the no. of elements");
Scanf("%d", &n);
for(i=1; i<=n; i++)
{
printf("Enter no %d", i+1);
Scanf("%f", &num[i]);
sum = sum + num[i];
}
avg = sum/n;
printf("Average = %f", avg);
Printf("sum = %.2f", sum);
getch();
}
```

Practical 06:

Aim: Factorial of a number using recursion.

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 1.

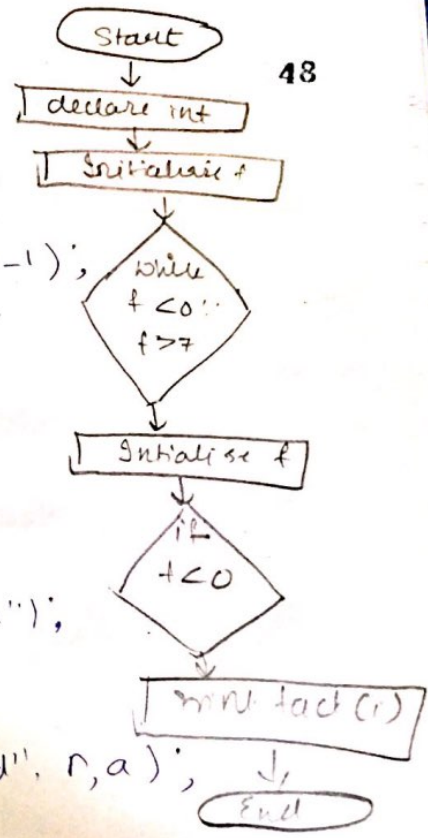
Step 4: use the while function to ensure that entered value is in the range 0 to 7.

Step 5: Print the value entered by the function in step 2.

Step 6: Terminate the program.

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 ();
}
```

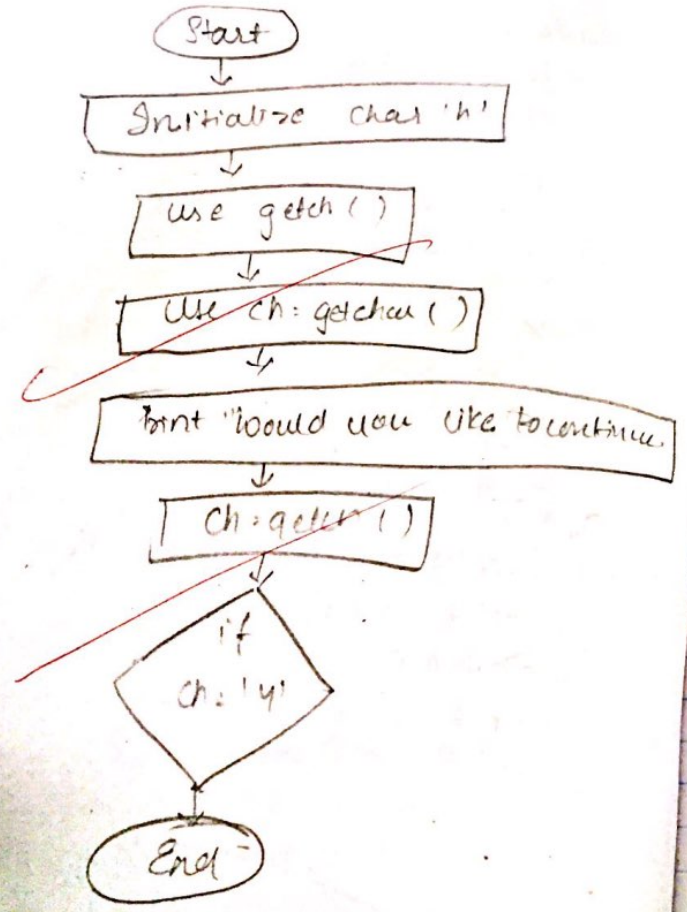


Output:

Enter a no to find the factorial: 90

Enter a no in range 0 to 7: 6

7:20



II] CNAP to show the use of get() function :

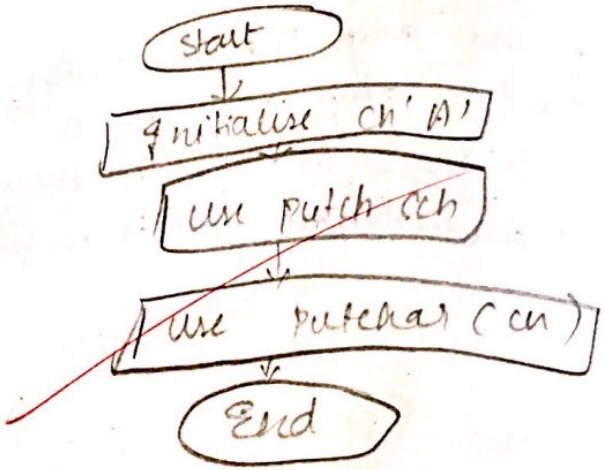
Step 1 : Start the turbo C application
Step 2 : Initialize the character variable 'ch'
Step 3 : use the getch() method to accept the character.
Step 4 : use the getchar() method to show an option 'y/n' in 'ch'
Step 5 : while ch == 'y', keep accepting value for ch.
Step 6 : use the getch() method to store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

Source code :-

```
#include <stdio.h>
#include <conio.h>
void main ()
{
   char ch: 'n'
   Clrscr ();
   printf (" Press any key to continue :");
   getch ();
  Printf ("Press any character");
   ch=getchar ();
 printf ("Would you like to continue(y/n)");
   ch: getche ()
while (A:'y')
{
  printf ("would you like to continue");
    ch: getche ()
  }
}
```

Output

```
Press any key to continue
Enter any character : A
Would you like to continue := y
Would you like to continue : n
```



(III) C.N.A.P. to Show the use of put() function

Step 1 : Start the turboc application,

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

Step 3 : use the putch() and putchar() function with ch as argument

Step 4 : Terminate the program

12.

→ Source code

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

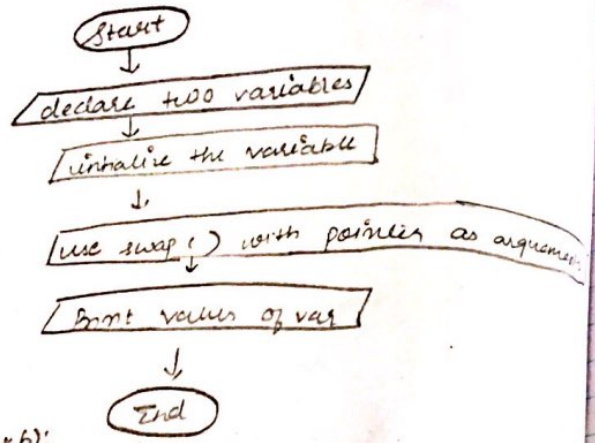
Conclusion: The factorial are the been done successfully

Ahmadi

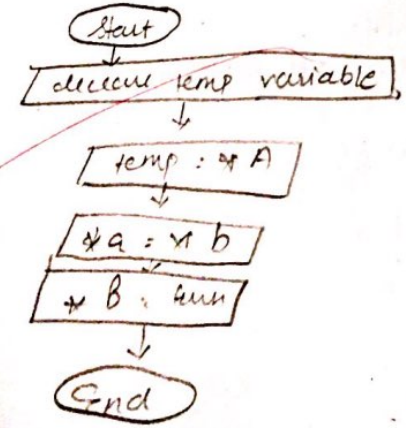
output

A

Flowchart



Swap (*a, *b):



Practical : 07

1] Swapping using pointers.

Aim

Algorithm

Step 1 : Start turbo c application.

Step 2 : Declare a f° prototype with two int-pointers as arguments before entering main ( )

Step 3 : Declare the variable and accept their value from the user. Print the respective values using printf ( )

Step 4 : Pass the address of the variables as arguments for the funct°

Step 5 : Print the respective value of the variable.

Step 6 : Using the basic swapping in the function definition but instead of normal variables, use pointers.

12.

Source code :-

```
void swap (int *m, int *n);
#include <conio.h>
#include <stdio.h>
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", x, y);
  swap(&x, &y);
  printf("The values after swapping are %d and %d respectively", x, y);
  getch();
}
void swap (int *m, int *n)
{
  temp = *m;
  *m = *n;
  *n = temp;
}
```

output :

Enter two nos to be swapped 12

2 4

The nos. before swapping are 12 and 2 4.

The nos after swapping are 2 4 & 12

(ii) SORTING OF ARRAYS USING POINTERS:-

Algorithm:-

Step 1: Initialize an integer i, j and temp variables

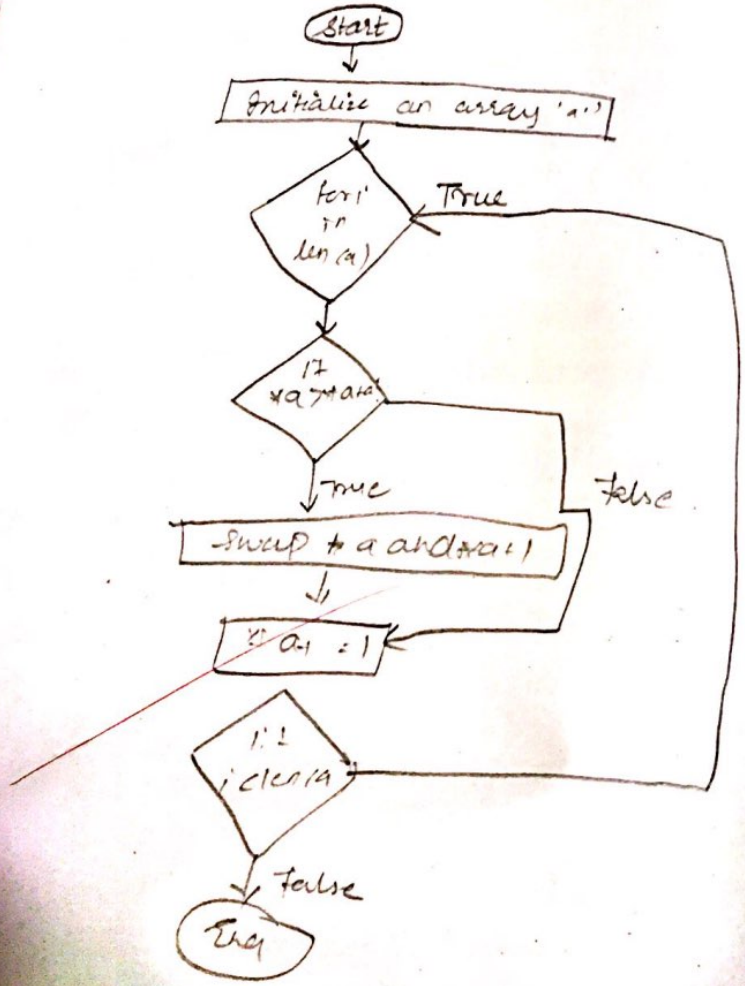
Step 2: Run a nested loop of i = 0 to len(a) and a j = 0 to len(a)

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

Step 4: Print the swapped array.

Step 5: Terminate the program.

Flowchart:



Source code

```
#include <stdio.h>
#include <conio.h>
void main()
int a[10], i, j, temp;
clrscr();
for (i=10; 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", a);
getch();
}
```

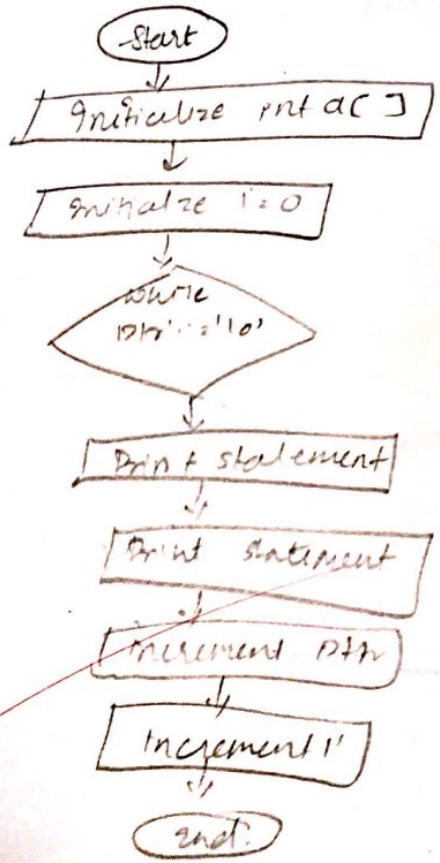
Output

Enter elements into the array:

6
7
8
2
9
10
11
5
10

{1, 2, 5, 6, 7, 8, 9, 10, 11} is the sorted array

Flowchart



Transversal of one-dimensional array using pointers:-

Algorithm:-

Step 1: Start Turbo C application

Step 2: Declare an integer array and a variable;

Step 3: Run a while loop with 0 to length of array.

Step 4: Print the default statement

Step 5: Increment pointer and i

Step 6: Terminate the program.

Conclusion: Thus, we have executed program using pointers successfully.

Ahmedi

Source code

```
#include <stdio.h>
#include <conio.h>
void main ()
{
int a[5] = { 7, 9, 4, 8, 2};
unit *ptr;
int i = 0;
ptr = &a[0];
while (*ptr != '\0')
{
Printf("\n\nThe address of a[%d] = %u",
        i, ptr);
Printf("\nThe value of a[%d] = %d",
        i, *ptr);
ptr++;
i++;
}
getch();
}
```

output

The value of a[1] = 9

The address of a[2] = 6550
The value of a[2] = 4

The address of a[3] = 6552
The value of a[3] = 8

The address of a[4] = 6554
The value of a[4] = 2

The address of a[0] = 6516
The value of a[0] = 7

The address of a[1] = 6518
The value of a[1] = 9

The address of a[2] = 6520
The value of [2] = 4

The address of a[3] = 6522
The value of a[3] = 8

The address of a[4] = 6524
The value of a[4] = 2

Practical: 08 Output-

:2 Enter id : 10
Enter cgpa : 8
Enter name : Esha
* * * * **
Id 10
Cgpa = 8
Name - Esha

Source code

```
#include<conio.h>
struct student
{
    int id, cpga;
    char name[10];
};
void main()
{
    struct student *p;
    clrscr();
    Printf("\n Enter id:");
    Scanf("%d", &p->id);
    Printf("\n Enter cpga:");
    Scanf("%d, &p->cpga");
    Printf("\n Enter name:");
fflush(stdin);
Scanf("%s", p->name);
Printf("\n**********\n");
Printf("\n Id =%d \n Cpga =%d \n Name =%s",
        p->id, p->cpga, p->name);
getch();
```

Q.2. Practical 08.

Algorithm.

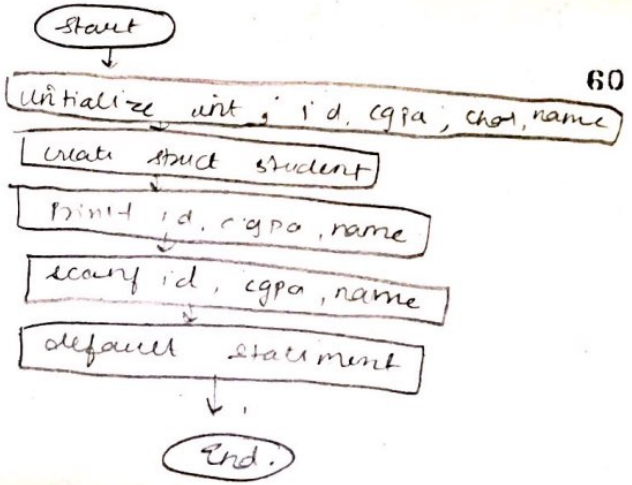
A] Create a single structure named 'student' that holds the following variable.

i) i'd (ii) CGPA (iii) Name

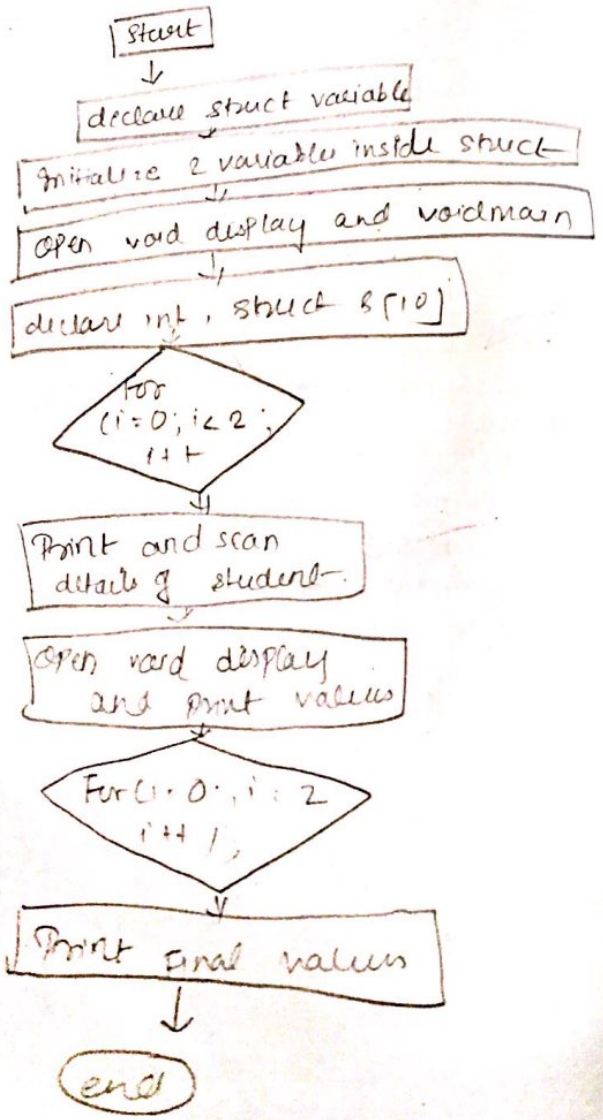
ALGORITHM

Step 1 : Start Turbo 'C' application.
Step 2 : Declare int i'd, cgpa and character 'name' of length 10.
Step 3 : Create the structure of name student.
Step 4 : Create Structure with pointer.
Step 5 : Print statements for entering id, name and cgpa using printf function
Step 6 : Store the info using scanf.
Step 7 : Print default statement.
Step 8 : Terminate the program.

Conclusion : Thus, we have successfully executed the program.



Flow chart [8] b



Practical : 08 [b]

Aim WAP which will demonstrate use of structure and function.

algorithm :

step 1 : Start turbo C application.

step 2 : Initialize the struct student with two name variable (int roll and char name [10])

step 3 : Now inside void main display and void main declare int & struct student s[10]

step 4 : use the for loop for entering details of student upto 2 students and not more than that.

step 5 : Print the details of students.

step 6 : Open void display again and print the value using for condition and print

step 7 : Terminate the program

:a

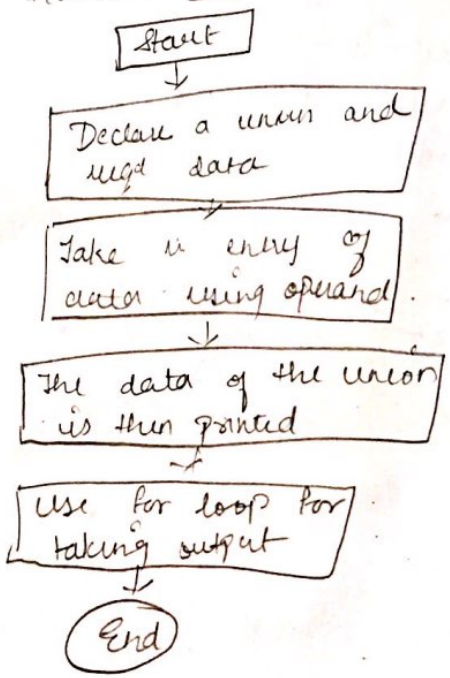
Code:

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

Output

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

Flowchart [C]



[C] Prac - 08 63

Aim: union

Algorithm:

Step 1 : Start the turbo C application

Step 2 : Use the union keyboard to declare the union of different datatypes

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

Step 4 : Now print all the data of union

Step 5 : Terminate the program.

:3

Code:

```
# include <stdio.h>
#include <conio.h>
    void main
    union student
{
    int rollno;
    char name[10], div, contactno[10]
      float percentage;
};
  void mark( )
{
    union student s;
    Printf("Enter the details')
    Scanf("%d, %s, %s, %s, %f", as rollno,
              name, div, contact no, percentage);
Printf("\n The details of the student are:');
Printf("Roll no %d; rollno);
Printf("\n Name %s, sname);
Printf("\n DIV %s, sdiv);
Printf("\n contact no %s, scontno);
Printf("\n Percentage %f. &percentage);
  getch( );
}
```

output

Enter the details

199

Esha

B

9922112212

76.90%.

The details of the student are:

Roll no: 199

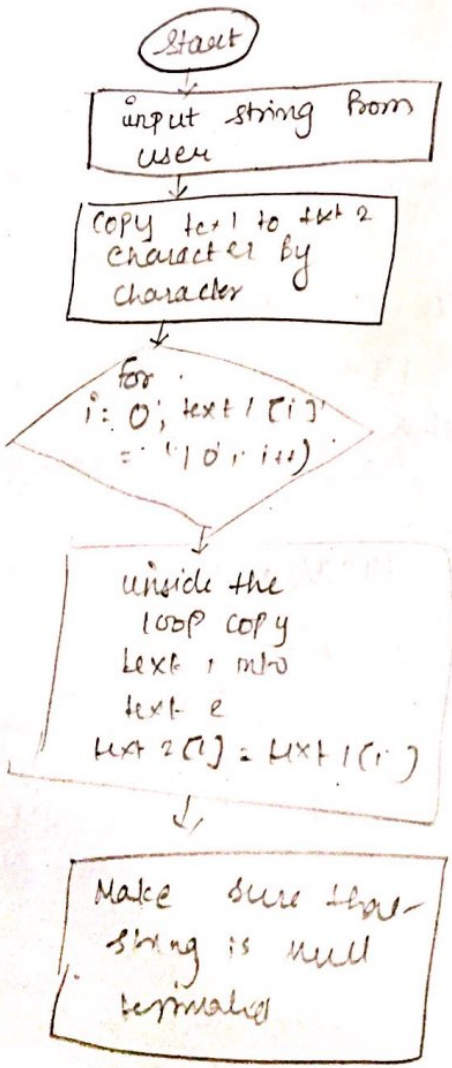
Name: Esha.

Div B

contact no: 9922112212

percentage: 76.90%.

9[A]



Pratical 09 (A)

Aim: WAP to copy one string into another string

Algorithm:

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

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

Step 3: Run a loop from 0 to end of string. The loop structure should be like
for (i=0; text 1 (i) ; = '\0'; i++) .

Step 4: Inside the loop for each character in text 1 copy to text 2. Say text 2 (i) = text 1(i)

Step 5: Finally after loop make sure the copied string end with Null character i.e. text 2 (i) = '\0';

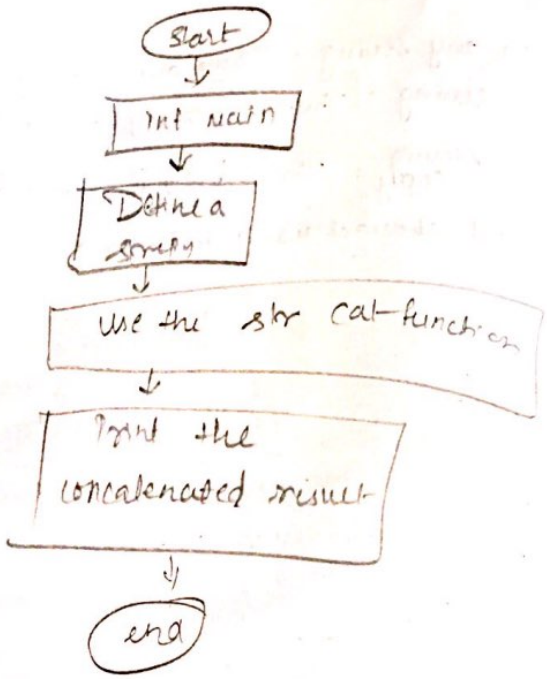
Q3

Code

```
# include <studio.h>
# define Max-Size 100

int main ()
{
    Char text 1 [MAX-SIZE];
    Char text 2 [MAX-SIZE];
    int i;
    print f (" Enter any string: ");
    gets (text 1);
    for (i=0) text 1 [i] != '0'; i++)
    {
      text 2 [i] = text 1 [i];
    }
  text 2 [i] = '10';
  print f (" First string = %. S\n", text 1);
  print f (" First string copy = %. S\n", text 2);
  print f (" Total Characters copied = %. d\n", i);

  return 0;
}
```

Output :

Enter any string : There are 7 days in a week
First string : There are 7 days in a week
First string copy : There are 7 days in a week
Total characters copied : 26.

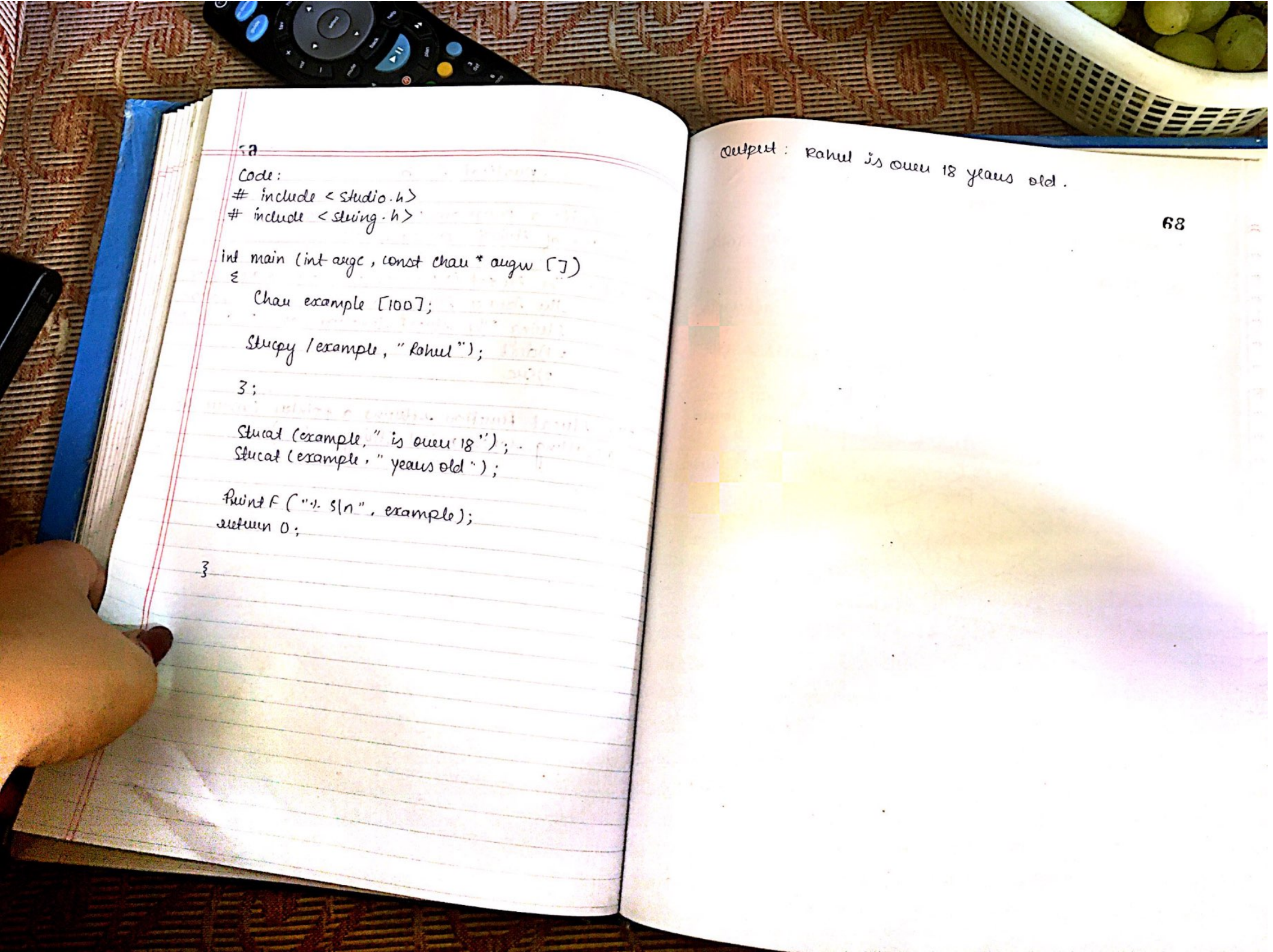


Practical 09 (b)

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 returns a pointer (where the resulting concatenated string resides)



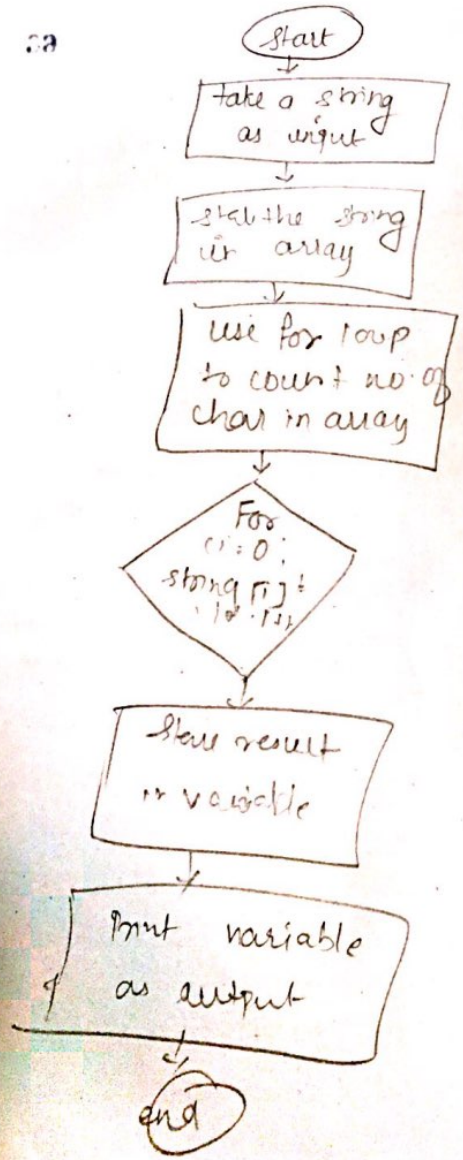
Ca

Code:

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

int main (int argc, const char * argv [])
{
   Char example [100];

   Strcpy (example, "Rahul");

   3;

   Strcat (example, " is over 18");
   Strcat (example, " years old");

Printf ("%s\n", example);
return 0;

}
```

output: Rahul is over 18 years old.

9(c)



Practical 09 (c) 69

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

Algorithm:

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

ca.

Code:

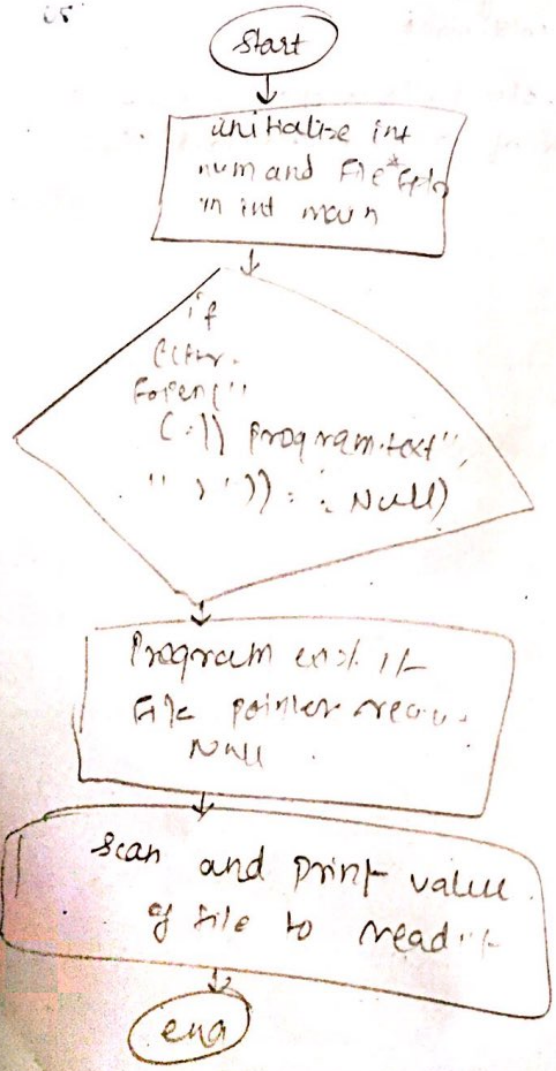
```
# include < stdio.h >
void main ()
{
    char string [50];
    int i, length = 0;

    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 it");
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 night = 18.

10(A)



Practical 10 (A) 71

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

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

f read() → Reads a Record from a file

f close() → closes a file

17

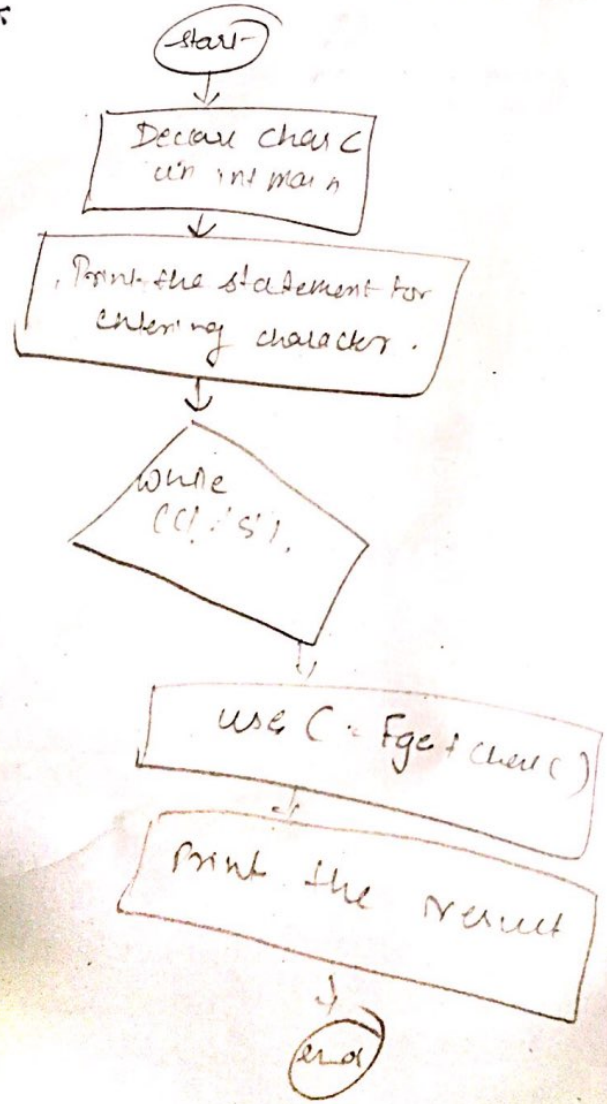
Code for reading from a text file / opening / closing
Text file is (.txt and its contents are : 87
88
89
90

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

if ((fptu = Fopen ("C:\\ program.txt", "r")) == NULL)
 {
   print F ("Error; opening file");

   exit (1);
 }
Fscan f (fptu, "%d" ; &num );
            are
printf ("value n = %d", num);
 Fclose (fptu);

return 0;
}
```

output:

Values are : 87
88
89
90

0(b)



Practical 10 (b) 73

Aim: WAP for Fgetc(), Fgetchar(), Fputchar() function

Algorithm / Description

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

Q5.

Code :

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

Output : Enter some character. Enter $ to exit ...

A

Entered character is : A

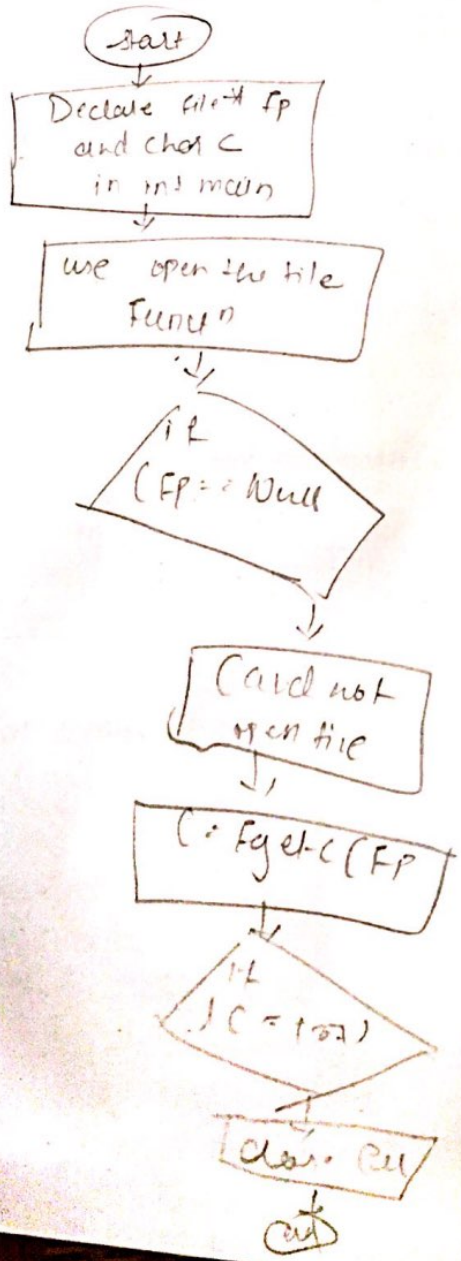
B

Entered character is : B

$

Entered character is $

(1)



fgetc() → used to read a character from a file Reads single character at a time in a program we use Fgetc() Function F getc(), where Fp = file pointer

Code:

```
include <stdio.h>
int main()
{
  File * Fp;
  Char c;
Printf ("opening file c in read mode");
  Fp = Fopen ("test.c", "r");
'F (Fp == Null)
{
  Printf ("Could not open file test.c");
  return 1;
}
Printf ("Reading the file test.c");
  While (1)
{
  c = Fgetc (Fp);
if (c = Fof)
  break;
  Printf ("%c", c);
}
```

```
Printf("Closing File test.c");
Fclose(Fp);
return 0;
}
```

output

```
opening the file test.c in read mode
Reading the file test.c
Hi, How are you?
closing the file test.c
```