**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**
**Ramapuram, Chennai- 600089.**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**Department of Computer Science & Engineering**

# LAB MANUAL

## 18CSS101J PROGRAMMING FOR PROBLEM SOLVING LAB

| | | |
|---|---|---|
| **CLASS** | **:** | B.Tech. [U.G] |
| **YEAR / SEM.** | **:** | I Year / II Semester |
| **SOFTWARE REQUIREMENT** | **:** | "C " |

**LIST OF EXPERIMENTS**

1. Algorithm, Flow Chart, Pseudocode
2. Input and Output Statements
3. Data Types
4. Operators and Expressions
5. Control Statements
6. Arrays – One Dimensional
7. Arrays – Multidimensional
8. Strings
9. Functions
10. Pointers
11. Structures and Unions
12. File Handling

| Ex.No: 1 | **Algorithm,Flowchart and Pseudocode** |
|---|---|
| | |

**Exercise 1(a):**

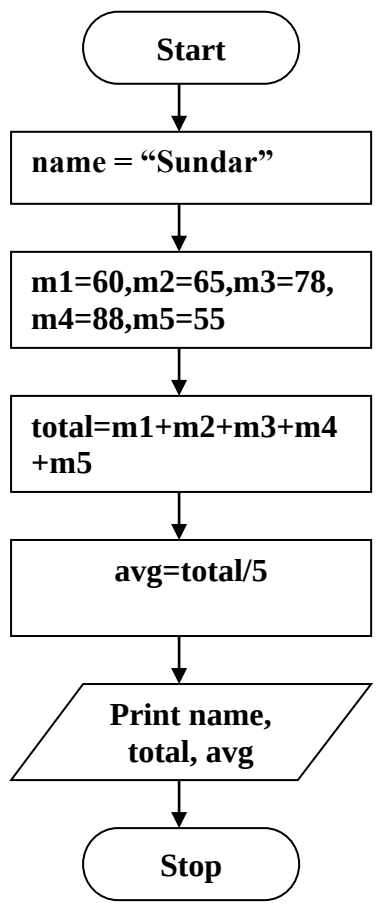
**Aim:**

To write a simple C program to find the total and average of a student's marks.

**Algorithm:**

**Step1:** Start the program.
**Step2:** Declare the required variables.
**Step3:** Assign the values to the declared variables.
**Step4:** Find total and average for the marks.
**Step5:** Display the output.
**Step6:** Stop the program.

**Flow Chart:**



**Pseudocode:**

**SET** name=”Sundar”,m1=60,m2=65,m3=78,m4=88,m5=55
**COMPUTE** total=m1+m2+m3+m4+m5
**COMPUTE** avg=total/5
**DISPLAY** name,total,avg

**Program:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int m1, m2, m3, m4, m5;
        int total;
        float avg;
        char name[15]={"sundar"};
        clrscr();
        m1 = 60;
        m2 = 65;
        m3 = 78;
        m4 = 88;
        m5 = 55;
        total = m1 + m2 + m3 + m4 + m5;
        avg = total / 5;
        printf("\nStudent Name : %s", name);
        printf("\nTotal Marks  : %d", total);
        printf("\nAverage Marks: %f", avg);
        getch();
}
```

**Exercise 1(b)**

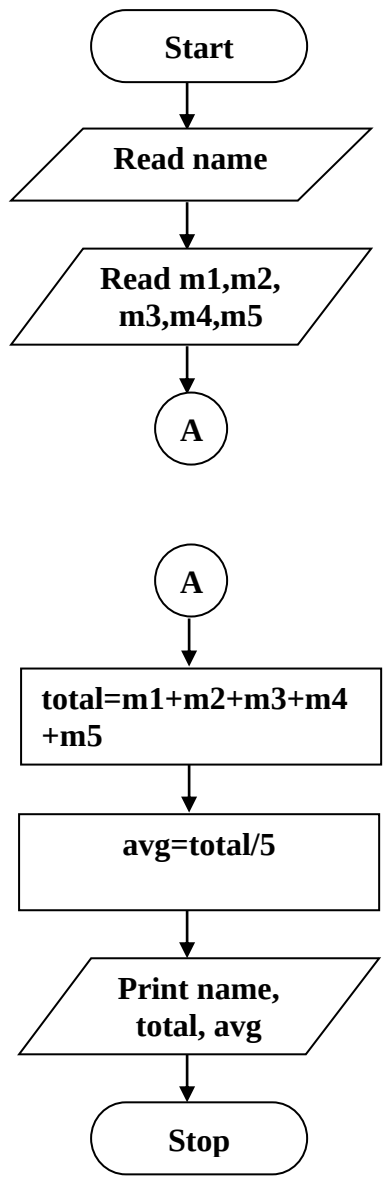
**Aim:**

To write a simple C program to find the total and average of a student's marks.

**Algorithm:**

**Step1:** Start the program.
**Step2:** Declare the required variables.
**Step3:** Get the values for the declared variables using scanf function.
**Step4:** Find total and average for the marks.
**Step5:** Display the output.
**Step6:** Stop the program.

**Flow chart:**




**Pseudocode:**

**INPUT** name,m1,m2,m3,m4,m5
**COMPUTE** total=m1+m2+m3+m4+m5
**COMPUTE** avg=total/5
**DISPLAY** name,total,avg

**Program:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int m1, m2, m3, m4, m5;
        int total;
        float avg;
        char name[15];
        clrscr();
        printf("\nEnter Student Name :");
        scanf("%s", name);
        printf("\nEnter Mark1: ");
        scanf("%d", &m1);
        printf("\nEnter Mark2: ");
        scanf("%d", &m2);
        printf("\nEnter Mark3: ");
        scanf("%d", &m3);
        printf("\nEnter Mark4: ");
        scanf("%d", &m4);
        printf("\nEnter Mark5: ");
        scanf("%d", &m5);
        total = m1 + m2 + m3 + m4 + m5;
        avg = total / 5;
        printf("\nStudent Name : %s", name);
        printf("\nTotal Marks  : %d", total);
        printf("\nAverage Marks: %f", avg);
        getch();
}
```

**Exercise 1(c):**

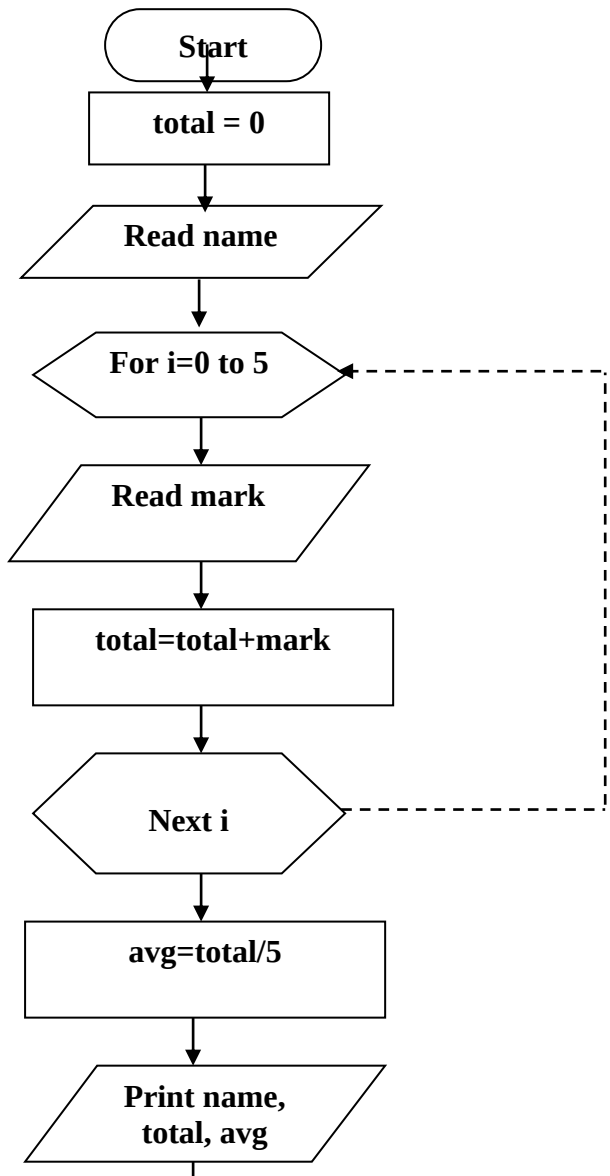
**Aim:**

To write a simple C program to find the total and average of a student's marks using 'for' loop.

**Algorithm:**

**Step1:** Start the program.
**Step2:** Declare the required variables.
**Step3:** Get the values for the declared variables using 'for' loop.
**Step4:** Find total and average for the marks.
**Step5:** Display the output.
**Step6:** Stop the program.

**Flow Chart:**



**Pseudocode:**

**SET** total=0
**INPUT** name
**FOR** i=0 to 5
    **INPUT** mark
    **COMPUTE** total=total+mark
**NEXT** i
**COMPUTE** avg=total/5
**DISPLAY** name, total, avg

**Program:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int mark;
        int total=0,i=0;
        float avg;
        char name[15];
        clrscr();
        printf("\nEnter Student Name :");
        scanf("%s", name);
        printf("\nEnter Marks for 5 Subjects... ");
        for(i=0; i<5; i++)
        {
                printf("\nMark%d: ", i+1);
                scanf("%d", &mark);
                total = total+mark;
        }
        avg = total / 5;
        printf("\nStudent Name : %s", name);
        printf("\nTotal Marks  : %d", total);
        printf("\nAverage Marks: %f", avg);
        getch();
}
```

**Exercise 1(d):**

**Aim:**

To write a simple C program to find the total and average of students marks using arrays.

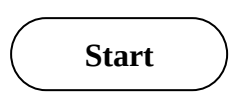
**Algorithm:**

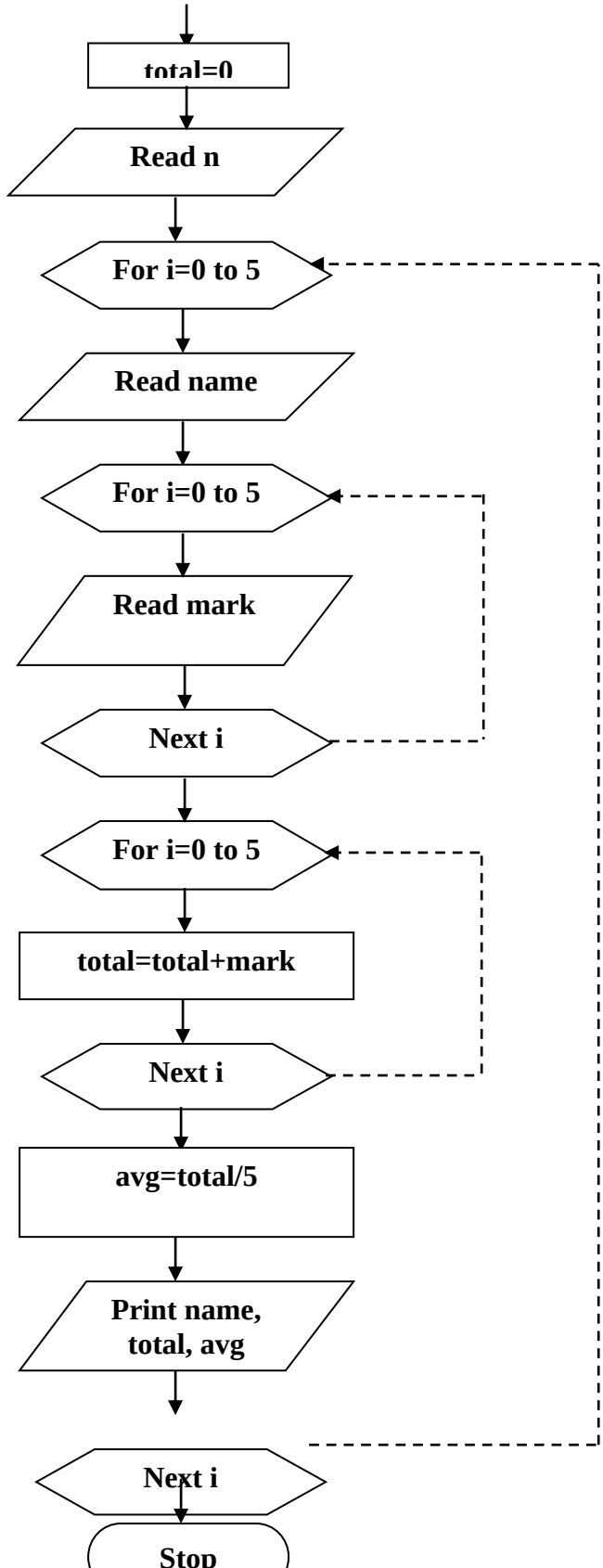
**Step1:** Start the program.
**Step2:** Declare the required variables.
**Step3:** Get the total number of students and store their names in an array.
**Step4:** Get the values for the declared variables for all students using 'for' loop.
**Step5:** Store their marks in another array.
**Step6:** Find total and average for the marks.
**Step7:** Display the output.
**Step8:** Stop the program.

**Pseudocode:**

```
SET total=0
INPUT number of students 'n'
FOR i=0 to 5
        INPUT name
        FOR i=0 to 5
                INPUT mark
        NEXT i
        FOR i=0 to 5
                COMPUTE total= total+mark
        NEXT i
COMPUTE avg=total/5
DISPLAY name, total, avg
```

**Flow chart:**




total=0
Read n
For i=0 to 5
Read name
For i=0 to 5
Read mark
Next i
For i=0 to 5
total=total+mark
Next i
avg=total/5
Print name,
total, avg
Next i
Stop

| Ex.No: 2 | Input , Output Statements |
|---|---|
| | |

**Aim:**

To Write a C Programs to

i) find the Area of Triangle,

ii) Find Area & circumference of circle,

iii) Convert the ASCII value of a character.

**Algorithm 1:**

1. Start the program
2. Read the three sides of triangle a, b and c
3. Calculate the area using the formula $\sqrt{s(s-a)(s-b)(s-c)}$
4. Till the value of s=(a+b+c)/2
5. display the area
6. Stop

**Algorithm2:**

1. Start the program
2. Declare the varaiables rad, area,ci
3. Get the values using scanf()
4. Calculate the Area of circle and circumference of a circle.
5. Stop

**Algorithm 3:**

1. Start the program
2. Declare the character
3. Get the characterusing scanf()
4. ASCII value of the character will be displayed.
5. Stop

**Program 1:**

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
  float area,s,a,b,c;
  clrscr();
  printf("Enter three sides of the Triangle for calculating Area\n");
  scanf("%f%f%f",&a,&b,&c);
  s=(a+b+c)/2;
  area=sqrt(s*(s-a)*(s-b)*(s-c));
  printf("\nThe Area of the Triangle is = %f",area);

  getch();
  return 0;
}
```

**Output 1:**

```
Enter the values of a, b and c
 3
 4
 5
 Area of Triangle = 6
```

**Program 2:**

```
#include<stdio.h>
 int main()
{
    int rad;
   float PI = 3.14, area, ci;
printf("\nEnter radius of circle: ");
  scanf("%d", &rad);
area = PI * rad * rad;
   printf("\nArea of circle : %f ", area);
ci = 2 * PI * rad;
   printf("\nCircumference : %f ", ci);
 return (0);
}
```

**Output2:**

```
Enter radius of a circle : 1
Area of circle : 3.14
Circumference  : 6.28
```

**Program 3:**

```
#include <stdio.h>
int main(){
char c;
printf("Enter a character: ");
scanf("%c",&c);
printf("ASCII value of %c = %d",c,c);
return 0;
}
```

**Output 3:**

```
Enter a character: G
ASCII value of G = 71
```

| Ex.No: 3 | DATA TYPES |
| --- | --- |
| | |

Aim:

To Write a C Programs to

i) find the Net Salaryof the employee

ii) Perform all arithmetic operations on 2 given values.

**Algorithm1:**

1. Start the program
2. Read the Basic Salary of the Employee
3. Calculate da = (basic_salry * 25)/100;
4. Calculate hra = (basic_salry * 15)/100;
5. Calculate gross_salry = basic_salry + da + hra;
6. Calculate pf = (gross_salry * 10)/100;
7. Calculate net_salry = gross_salry - pf;
8. display theNet Salaryof the employee
9. Stop

**Algorithm 2:**

1. Start the program
2. Assign the values of a & b
3. Perform all the arithmetic operations such as addition, subtraction, multiplication, Division, modulo operation and increment and decrement operations.
4. Display the values.
5. Stop

**Program 1:**

```
#include <stdio.h>
#include<conio.h>
main()
{
clrscr();
floatbasic_salry, da, hra, pf, gross_salry, net_salry;
printf("\nEnter the Basic Salary of the Employee: Rs. = ");
scanf("%f" , &basic_salry);
da = (basic_salry * 25)/100;
hra = (basic_salry * 15)/100;
gross_salry = basic_salry + da + hra;
pf = (gross_salry * 10)/100;
net_salry = gross_salry - pf;
printf("\n\nNet Salary: Rs. = %.2f", net_salry);
getch();
}
```

**Output 1:**

```
Enter the Basic Salary of the Employee: Rs.5000.00
Net Salary: Rs. = 6300.00
```

**Program 2:**

```
#include <stdio.h>
main()
{
int a = 21;
int b = 10;
int c ;
   c = a + b;
printf("Line 1 - Value of c is %d\n", c );
   c = a - b;
printf("Line 2 - Value of c is %d\n", c );
   c = a * b;
printf("Line 3 - Value of c is %d\n", c )
   c = a / b;
printf("Line 4 - Value of c is %d\n", c );
```

```
   c = a % b;
printf("Line 5 - Value of c is %d\n", c );
   c = a++;
printf("Line 6 - Value of c is %d\n", c );
   c = a--;
printf("Line 7 - Value of c is %d\n", c );
}
```

**Output 2:**

```
Line 1 - Value of c is 31
Line 2 - Value of c is 11
Line 3 - Value of c is 210
Line 4 - Value of c is 2
Line 5 - Value of c is 1
Line 6 - Value of c is 21
Line 7 - Value of c is 22
```

| Ex.No: 4 | Operators and Expressions |
|---|---|
| | |

Write a program to take input of name, rollno and marks obtained by a student in 4 subjects of 100 marks each and display the name, rollno with percentage score secured.

Algorithm:

1. Start
2. Define variables: name, rollno, sub1, sub2, sub3, sub4, sum, score
3. Take input from keyboard for all the input variables
4. Calculate the sum of marks of 4 subjects and also calculate the percentage score as:

sum = sub1 + sub2 + sub3 + sub4; score = (sum/400) * 100

5. Display the name, roll number and percentage score.
6. Stop

PROGRAM

```
#include<stdio.h>
 #include<conio.h>
void main()
{
char name[20]; int rollno;
float sub1, sub2, sub3, sub4, , sum, score; printf("Enter name of student: "); scanf(“%s”,&name[]);
printf ("\n Enter Roll Number: "); scanf("%d", &rollno);
printf ("\n Enter Marks in 4 Subjects:\n"); scanf("%f%f%f%f", &sub1, &sub2, &sub3, &sub4);
sum=sub1+sub2+sub3+sub4;
score = (sum/500)*100;
printf("\n Name of student: %s", name[]); printf("\n Roll Number: %d", rollno);
printf ("\nPercentage score secured: %2.2f%c", score,'%'); getch();
}
```

Output:

```
Enter name of student: Ajit Singh Roll Number: 25
Enter Marks in 4 Subjects: 50
75
85
62
Name of student: Ajit Singh Roll Number: 25
Percentage score secured: 68.00%
```

| Ex.No: 5(a) | Control Statements |
| --- | --- |
| | |

**Aim:**

To Write a C Programs to find the Largest among three numbers.

**Algorithm 1:**

1. Start the program
2. Read the three values for a, b and c
3. Compare a > b and a>c is true display a is greater
4. compare b >c is true display b is largest.
5. display c is largest number
6. Stop

**Program:**

```
#include<studio.h>
#include<conio.h>
int main()
{
float a, b, c;
printf("Enter three numbers: ");
scanf("%f %f %f", &a, &b, &c);
if(a>=b && a>=c)
printf("Largest number = %.2f", a);
else if(b>=a && b>=c)
printf("Largest number = %.2f", b);
else
printf("Largest number = %.2f", c);
return 0;
}
```

**Output:**

```
Enter three numbers: 12.2
13.452
10.193
Largest number = 13.45
```

| Ex.No: 5(b) | Control Statements-Use of for, while and do…while loops |
|---|---|
| | |

**Aim:**

To Write a C Programs to

i) To check the given number is palindrome or not

ii) To find Fibonacci series

iii) To find the factorial of the given number.

**iv) To add all the numbers entered by a user until user enters 0.**

**Algorithm 1:**

1. Start the program
2. Declare the variables n, rem,temp and initialize reverse=0;
3. Get the integer values using scanf()
4. Give condition in while loop if the condition is true then the following statements will be executed
5. rem=temp%10;
6. reverse=reverse*10+rem;
7. temp/=10;
8. If reverse = number then it is a palindrome
9. Stop

**Algorithm 2:**

1. Start the program
2. Declare and initialise the variables
3. Enter the range of numbers
4. Give the condition in for loop
5. Display the Fibonacci series

**Algorithm 3:**

1. Start the program
2. Declare the variablesintnumber,factorial;
3. Enter the number
4. Check for the condition using while loop
5. If the condition is true get the factorial of a number
6. Stop

**Algorithm 4:**

1. Start the program
2. Declare the variables to **add all the numbers entered by a user until user enters 0.**
3. Enter the number
4. Check the condition in do while loop
5. If the condition is true the numbers will be added until the user enters 0

**Program 1:**

```
#include <stdio.h>
int main()
{
int n, reverse=0, rem,temp;
printf("Enter an integer: ");
scanf("%d", &n);
temp=n;
while(temp!=0)
  {
rem=temp%10;
reverse=reverse*10+rem;
temp/=10;
  }
if(reverse==n)
printf("%d is a palindrome.",n);
else
printf("%d is not a palindrome.",n);
return 0;
}
```

**Output 1:**

```
Enter an integer: 12321
12321 is a palindrome.
```

**Program 2**

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

```
   int k,r;
  long int i=0l,j=1,f;
  printf("Enter the number range:");
   scanf("%d",&r);
   printf("FIBONACCI SERIES: ");
   printf("%ld %ld",i,j);
   for(k=2;k<r;k++)
{
       f=i+j;
       i=j;
       j=f;
       printf(" %ld",j);
   }
  return 0;
}
```

**Output 2:**

Enter the number range: 15

FIBONACCI SERIES: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

**Program 3:**

```
#include <stdio.h>
int main()
{
intnumber,factorial;
printf("Enter a number.\n");
scanf("%d",&number);
factorial=1;
while (number>0)
{
factorial=factorial*number;
         --number;
}
printf("Factorial=%d",factorial);
return 0;
}
```

**Output 3:**

Enter a number.
5
Factorial=120

**Program 4:**

```
#include <stdio.h>
int main()
{
int sum=0,num;
do
   {
printf("Enter a number\n");
scanf("%d",&num);
sum+=num;
   }
while(num!=0);
printf("sum=%d",sum);
return 0;
}
```

**Output 4:**

```
Enter a number
3
Enter a number
-2
Enter a number
0
sum=1
```

| Ex.No: 6 | Arrays |
| --- | --- |
| | |

**Aim:**

To Write a C Programs to

i) find the largest and smallest number in the given array

ii) To arrange the contents of array in ascending order

iii) To calculate sum and average of array contents

**Algorithm1:**

1. Start the program
2. Read the input value to n
3. Repeat for i=0 to n
4. Read the input values to array a[i]
5. Check the condition using if loop
6. Stop the Program

**Algorithm 2:**

1. Start the program
2. Read the input value to n
3. Set loop i from 1 to size
4. Read a[i]
5. Check the value of number[i] is greater then number[2]
6. Swap number[i], number[j]
7. Print the number
8. Stop the program

**Algorithm 3:**

1. Start the program
2. Read input n
3. Read the array elements
4. Compute sum of positive numbers
5. Compute sum of negative numbers
6. Compute Average of positive and negative numbers
7. Stop the program

**Program 1:**

```
#include<stdio.h>
int main()
{
  int a[50],size,i,big,small;
  printf("\nEnter the size of the array: ");
  scanf("%d",&size);
  printf("\nEnter %d elements in to the array: ", size);
  for(i=0;i<size;i++)
  scanf("%d",&a[i]);
  big=a[0];
  for(i=1;i<size;i++)
{
     if(big<a[i])
          big=a[i];
  }
  printf("Largest element: %d",big);
  small=a[0];
  for(i=1;i<size;i++){
     if(small>a[i])
          small=a[i];
  }
  printf("Smallest element: %d",small);
  return 0;
}
```

**Output 1:**

```
Enter the size of the array: 4
Enter 4 elements in to the array: 2 7 8 1
Largest element: 8
Smallest element: 1
```

**Program 2:**

```
#include <stdio.h>
void main()
{
inti, j, a, n, number[30];
printf("Enter the value of N \n");
scanf("%d", &n);
printf("Enter the numbers \n");
for (i = 0; i< n; ++i)
scanf("%d", &number[i]);
for (i = 0; i< n; ++i)
   {
for (j = i + 1; j < n; ++j)
      {
if (number[i] > number[j])
          {
             a =  number[i];
number[i] = number[j];
number[j] = a;
          }
       }
   }
printf("The numbers arranged in ascending order are given below \n");
for (i = 0; i< n; ++i)
printf("%d\n", number[i]);
}
```

**Output 2:**

```
Enter the value of N
6
Enter the numbers
3
78
90
456
780
200
The numbers arranged in ascending order are given below
3
78
90
200
```

456
780

**Program 3:**

```
#include  <stdio.h>
#define MAXSIZE 10
void main()
{
int array[MAXSIZE];
inti, num, negative_sum = 0, positive_sum = 0;
float total = 0.0, average;

printf ("Enter the value of N \n");
scanf("%d", &num);
printf("Enter %d numbers (-ve, +ve and zero) \n", num);
for (i = 0; i<num; i++)
   {
scanf("%d", &array[i]);
   }
printf("Input array elements \n");
for (i = 0; i<num; i++)
   {
printf("%+3d\n", array[i]);
   }
for (i = 0; i<num; i++)
{
if (array[i] < 0)
       {
negative_sum = negative_sum + array[i];
       }
else if (array[i] > 0)
       {
positive_sum = positive_sum + array[i];
       }
else if (array[i] == 0)
       {
          ;
       }
total = total + array[i] ;
   }
average = total / num;
printf("\n Sum of all negative numbers =  %d\n", negative_sum);
printf("Sum of all positive numbers =  %d\n", positive_sum);
printf("\n Average of all input numbers =  %.2f\n", average);
```

}

**Output 3:**

```
Enter the value of N
10
Enter 10 numbers (-ve, +ve and zero)
-8
9
-100
-80
90
45
-23
-1
0
16
Input array elements
 -8
 +9
-100
-80
+90
+45
-23
 -1
 +0
+16

Sum of all negative numbers =  -212
Sum of all positive numbers =  160

Average of all input numbers =  -5.20
```

| Ex.No: 7 | Mulitidimensional Array |
| --- | --- |
| | |

**Aim:**

To Write a C Programs to implement following matrix operation.

(i) Matrix addition

(ii) Matrix multiplication

**Algorithm 1 :**

1. Read the elements of first matrix A
2. Read the elements of second matrix B
3. Add the elements of first matrix with second matrix and store in matrix C
4. display the matrix C.

**Algorithm 2:**

1. Read the elements of first matrix A
2. Read the elements of second matrix B
3. Multiply the elements of first matrix with second matrix and store in matrix C
4. display the matrix C.

**Program 1:**

```
#include <stdio.h>
  int main()
{
int m, n, c, d, first[10][10], second[10][10], sum[10][10];
printf("Enter the number of rows and columns of matrix\n");
scanf("%d%d", &m, &n);
printf("Enter the elements of first matrix\n");
for (c = 0; c < m; c++)
{
for (d = 0; d < n; d++)
{
scanf("%d", &first[c][d]);
}
}
printf("Enter the elements of second matrix\n");
for (c = 0; c < m; c++)
{
for (d = 0 ; d < n; d++)
{
scanf("%d", &second[c][d]);
}
}
printf("Sum of entered matrices:-\n");
  for (c = 0; c < m; c++)
{
for (d = 0 ; d < n; d++)
{
sum[c][d] = first[c][d] + second[c][d]; printf("%d\t", sum[c][d]);
}
printf("\n");
}
return 0;
}
```

**Output :**

```
 Enter the number of rows and columns of first matrix
3
3
Enter the elements of first matrix
```

```
1 2 0
0 1 1
2 0 1

Enter the elements of second matrix
1 1 2
2 1 1
1 2 1
Sum of entered matrices:-
    1   3  2
    2   2  2
    3   2  2
```

**Program 2:**

Matrix Multipliction:

```
#include <stdio.h>

int main()
{
int m, n, p, q, c, d, k, sum = 0;
int first[10][10], second[10][10], multiply[10][10];

printf("Enter the number of rows and columns of first matrix\n");
scanf("%d%d", &m, &n);
printf("Enter the elements of first matrix\n");

for (  c = 0 ; c < m ; c++ )
for ( d = 0 ; d < n ; d++ )
scanf("%d", &first[c][d]);

printf("Enter the number of rows and columns of second matrix\n");
scanf("%d%d", &p, &q);

if ( n != p )
printf("Matrices with entered orders can't be multiplied with each other.\n");
else
  {
printf("Enter the elements of second matrix\n");

for ( c = 0 ; c < p ; c++ )
```

```
for ( d = 0 ; d < q ; d++ )
scanf("%d", &second[c][d]);

for ( c = 0 ; c < m ; c++ )
   {
for ( d = 0 ; d < q ; d++ )
     {
for ( k = 0 ; k < p ; k++ )
       {
sum = sum + first[c][k]*second[k][d];
       }

multiply[c][d] = sum;
sum = 0;
     }
   }

printf("Product of entered matrices:-\n");

for ( c = 0 ; c < m ; c++ )
   {
for ( d = 0 ; d < q ; d++ )
printf("%d\t", multiply[c][d]);

printf("\n");
   }
  }

return 0;
}
```

**Output:**

```
Enter the number of rows and columns of first matrix
3
3
Enter the elements of first matrix
1 2 0
0 1 1
2 0 1
Enter the number of rows and columns of second matrix
```

3
3

Enter the elements of second matrix
1 1 2
2 1 1
1 2 1

Product of entered matrices:-
5 6 3
3 3 2
3 4 5

| Ex.No: 8 | Strings |
| --- | --- |
| | |

```
/*Use built in function- find total number of words*/
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char s[200];
int count=0,i;
clrscr();
printf("Enter the string:");
scanf("%[^\n]s",&s);
for(i=0;s[i]!='\0';i++)
{
if (s[i]==' ')
count++;
}
printf("Number of Words in given String are:%d\n",count+1);
getch();
}
```

OUTPUT:

```
Enter the string:abc abcd
Number of Words in given String are:2
```

```
/* To capitalize the first word of each sentence.*/
#include<stdio.h>
#include<conio.h>
#define MAX 100
void main()
{
char str[MAX]={0};
int i;
clrscr();
printf("Enter a string: ");
scanf("%[^\n]s",str);
for(i=0;str[i]!='\0';i++)
{
if(i==0)
{
if ((str[i]>='a' && str[i]<='z'))
str[i]=str[i]-32;
continue;
}
if(str[i]==' ')
{
i++;
if(str[i]>='a' && str[i]<='z')
{
str[i]=str[i]-32;
continue;
}
}
else
{
if(str[i]>='A' && str[i]<='Z')
str[i]=str[i]+32;
}
}
printf("Capitalize string:%s\n",str);
getch();
}
```

OUTPUT

```
Enter a string:hello
Capitalize string:HELLO
```

```
/*To replace a given word with another word*/
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
char *rw(const char *s,const char *oldw,const char *neww)
{
char *result;
inti,cnt=0;
intnewwlen=strlen(neww);
intoldwlen=strlen(oldw);
for(i=0;s[i]!='\0';i++)
{
if(strstr(&s[i],oldw)==&s[i])
{
cnt++;
i+=oldwlen-1;
}
}
result=(char *)malloc(i+cnt *(newwlen-oldwlen)+1);
i=0;
while(*s)
{
if(strstr(s,oldw)==s)
{
strcpy(&result[i],neww);
i+=newwlen;
s+=oldwlen;
}
else
result[i++]=*s++;
}
result[i]='\0';
return result;
}
void main()
{
charstr[]="xxforxx";
char c[]="xx";
char d[]="Geeks";
char *result=NULL;
clrscr();
printf("Old string:%sn ",str);
```

```
result=rw(str,c,d);
printf(" New string:%sn",result);
free(result);getch();}
```

OUTPUT:
Old string:hello
New string:welcome

| Ex.No: 9(a) | Functions |
| --- | --- |
| | |

**Aim:**

To Write a C Programs using functions to implement

(i) To multiply two numbers

(ii) To find the maximum of two numbers

**Algorithm 1 :**

1. Create a function mult() with the parameters
2. Enter the input 2 numbers to be multiplied
3. Enter the values to be multiplied
4. Product of two numbers is got using mult() function

**Algorithm 2:**

1. Create a function max() with the parameters
2. Enter the input 2 numbers.
3. Maximum of two numbers is got by max() function

**Program 1:**

```
#include <stdio.h>
int mult ( int x, int y );
int main()
{
int x;
int y;

printf( "Please input two numbers to be multiplied: " );
scanf( "%d", &x );
scanf( "%d", &y );
printf( "The product of your two numbers is %d\n", mult( x, y ) );
getchar();
}

int mult (int x, int y)
{
return x * y;
}
```

**Output:**

```
Please input two numbers to be multiplied:
4
2
The product of two numbers is 8
```

## Program 2:

```
#include <stdio.h>
int max(int num1, int num2);
int main ()
{
int a = 100;
int b = 200;
int ret;
ret = max(a, b);
printf( "Max value is : %d\n", ret );
return 0;
}
```

```
int max(int num1, int num2)
{
int result;
if (num1 > num2)
{
result = num1;
}
else
{
result = num2;
}
return result;
}
```

**Output:**

Maximum value is 200.

| Ex.No: 9(b) | Recursion Function |
| --- | --- |
| | |

**Aim:**

To Write a C Programs to implement Recursion

**Algorithm 1:**

1. Declare the function sum()
2. Enter the positive integer
3. Perform summation using sum() function
4. Sum() is called recursively.

**Algorithm 2:**

1. Declare the function Fact()
2. Enter a positive Integer
3. Calculate the factorial using Fact() function
4. Fact() is called recursively

**Program 1:**

```
#include <stdio.h>
int sum(int n);
int main()
{
intnum,add;
printf("Enter a positive integer:\n");
scanf("%d",&num);
add=sum(num);
printf("sum=%d",add);
}
int sum(int n){
if(n==0)
return n;
else
returnn+sum(n-1);    /*self call  to function sum() */
}
```

**Output:**

```
Enter a positive integer:
5
15
```

**Program 2:**

```
#include<stdio.h>
int fact(int);
int main(){
  int num,f;
  printf("\nEnter a number: ");
  scanf("%d",&num);
  f=fact(num);
  printf("\nFactorial of %d is: %d",num,f);
  return 0;
}

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

**Output:**

Enter a  number :  6
Factorial of 6 is 720

| Ex.No: 10 | POINTERS |
|---|---|
| | |

**Aim:**

To Write C Programs to illustrate the use of pointers.

**Algorithm 1 :**

1. Declare the pointer variables.
2. Get the Address of the variables
3. Get the value for variables.
3.Print the Address and content of the Variables for different inputs.

**Algorithm 2 :**

1. Declare the structure and pointer variables.
2. Get the input for the variables
3. Print the values of variables using a loop.

**Program 1:**

```
#include <stdio.h>
int main()
{
int* pc;  int c;
c=22;
printf("Address of c:%d\n",&c);
printf("Value of c:%d\n\n",c);
pc=&c;
printf("Address of pointer pc:%d\n",pc);
printf("Content of pointer pc:%d\n\n",*pc);
c=11;
printf("Address of pointer pc:%d\n",pc);
printf("Content of pointer pc:%d\n\n",*pc);
*pc=2;
printf("Address of c:%d\n",&c); printf("Value of c:%d\n\n",c);
return 0;
}
```

**Output:**

```
Address of c: 2686784
Value of c: 22
Address of pointer pc: 2686784 Content of pointer pc: 22
Address of pointer pc: 2686784 Content of pointer pc: 11
Address of c: 2686784
Value of c: 2
```

**Program 2:**

```
#include<stdio.h>
struct invent
{
char  *name[20];
int number;
float price;
};
main()
```

```
{
struct invent product[3],*ptr;
printf(“INPUT\n\n”);
for(ptr=product;ptr<product+3;;ptr++)
scanf(“%s %d %f”,ptr->name,&ptr->number,&ptr->price);
printf(“\nOUTPUT\n\n”);
ptr=product;
while(ptr<product+3)
{
printf(“%-20s %5d %10.2f\n”,
        ptr->name,
        ptr->number,
        ptr->price);
ptr++;
}
}
```

**OUTPUT**

```
INPUT
Washing_machine     5       7500
Electric_iron       12      350
Two_in_one          7       1250

OUTPUT

Washing_machine     5       7500.00
Electric_iron       12      350.00
Two_in_one          7       1250.00
```

| Ex.No: 11 | STRUCTURES AND UNIONS |
|---|---|
| | |

**Program 1:** Write a C program to create, declare and initialize structure.

**Code:**

```
#include <stdio.h>
/*structure declaration*/ struct
employee{
        char name[30]; int
        empId; float salary;
};

int main()
{
        /*declare and initialization of structure variable*/ struct employee
        emp={"Anil",201,80000.00};


        printf("\n Name: %s"                    ,emp.name);
        printf("\n Id: %d"                      ,emp.empId);
        printf("\n Salary: %f\n",emp.salary); return 0;
}
```

**Program 2:** Write a program to store information of 5 students in structure and display it.

**Code:**

```
#include<stdio.h> struct
student
{
        char name[30]; int
        roll; float marks;
} s[5];
int main( )
{
        int i;
        printf(“Information of students:”);
```

```
        for (i=0; i<5; ++i)
        {
                s[i].roll =i+1;
                printf("\n Roll number %d, \n", s[i].roll); printf("Enter name:");
                scanf("%s", s[i].name); printf("Enter
                marks:"); scanf("%f", &s[i].marks);
        }
printf("\n Displaying Information:\n"); for(i=0;i<10;++i)
{
        printf("\n Roll number:%d \n", i+1); printf("Name:");
        puts(s[i].name);
        printf("\n Marks:%.1f", s[i].marks);
}
return 0;
}
```

**Program 3:** Write a program to declare, initialize an UNION.

```
#include <stdio.h>
// union declaration union
pack{
char a; int
        b;
double c;
};
int main()
{
        pack p; //union object/variable declaration printf("\nOccupied
        size by union pack:
%d",sizeof(pack));
        // assign value to each member one by one other it will replace last value
        p.a='A';
        printf("\nValue of a:%c",p.a); p.b=10;
        printf("\nValue of b:%d",p.b);
        p.c=12345.6790;
        printf("\nValue of c:%f",p.c);
        // see, what will happen? if u will assign values together
        p.a='A'; p.b=10;
        p.c=12345.6790;
        // here the last value of p.c will be accessed by all members
        printf("\nValue of a:%c, b:%d, c:%f",p.a,p.b,p.c); return 0;
}
```

| Ex.No: 12 | FILE HANDLING |
|---|---|
| | |

.

Program 1: Write a program to create a file called emp.rec and store information about a person, in terms of his name, age and salary.

Code:

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

FILE *fptr; char name[20]; int  age; float salary;
/*          open for writing */
fptr = fopen("emp.rec", "w"); if (fptr == NULL)
{
printf("File does not exists \n"); return;
}
printf("Enter the name \n"); scanf("%s", name);
fprintf(fptr, "Name        = %s\n", name);
printf("Enter the age\n"); scanf("%d", &age);
fprintf(fptr, "Age         = %d\n", age);
printf("Enter the salary\n"); scanf("%f", &salary);
fprintf(fptr, "Salary      = %.2f\n", salary);
fclose(fptr);
}
```

Program 2: Write a program to illustrate how a file stored on the disk is read.

Code:

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

FILE *fptr;
char filename[15]; char ch;
printf("Enter the filename to be opened \n");
scanf("%s", filename);
/*          open the file for reading */
fptr = fopen(filename, "r"); if (fptr == NULL)
{
```

```
printf("Cannot open file \n"); exit(0);
}
ch = fgetc(fptr); while (ch != EOF)
{
printf ("%c", ch); ch = fgetc(fptr);
}
fclose(fptr);
}
```