```
float a,b,c,r1,r2,disc,real_part,img_part;
 printf("Enter the coefficients of
quadratic equation");
 scanf("%f%f%f",&a,&b,&c);
 if(a==0)
 {
  printf("The equation is linear");
  return 1;
 }
 else
    {
     disc=(b*b)-4*a*c;
     if(disc==0)
     {
       printf("Real and equal roots\n");
       r1=r2=-b/(2*a);
       printf("The roots are r1= %f and r2=
%f\n",r1,r2);
     }
     else if(disc>0)
     {
      printf("Real and distinct roots\n");
      r1=(-b+sqrt(disc))/(2*a);
      r2=(-b-sqrt(disc))/(2*a);
      printf("The roots are r1= %f and r2=
%f\n",r1,r2);
     }
     else
     {
      printf("Imaginary roots\n");
      real_part=-b/(2*a);
      img_part= sqrt(-disc)/(2*a);
      printf("The roots are \n");
      printf("r1= %f +i
%f\n",real_part,img_part);
      printf("r2= %f -i
%f\n",real_part,img_part);
     }
  }
}
```

6:41 pm

```
length = str_length(str);

// Print the length of the string
printf("The length of %s is %d\n",str,length);

// Accept two strings to compare
printf("\nEnter two strings for string compare :");
scanf("%s%s",str1,str2);



// Invoke string compare function to compare the str1 and str2 strings
comp_res=str_compare(str1,str2);

if (comp_res < 0)
{
        printf("String \"%s\" is less than string \"%s\"\n",str1,str2);
}
else if (comp_res == 0)
{
        printf("String \"%s\" is same as string \"%s\"\n",str1,str2);
}
else
{
```

```
        // Accept two strings for string concatenation
        printf("\nEnter two strings for string concatenation :");
        scanf("%s%s",str_des,str_src);

        // Invoke string concatenation function
        str_concat(str_des,str_src);

        // Print the concatenated string
        printf("The string after concatenation is \"%s\"\n", str_des);

        return 1;
}

int str_length(char s[])
{
        int i;
        for(i=0;s[i]!='\0';i++);
        return i;
}


int str_compare(char s1[], char s2[])
{
        int i,j;
        for(i=0,j=0;(s1[i] != '\0' && s2[j] != '\0');i++,j++)
        {
                if (s1[i] != s2[j])
                {
                        return (s1[i] - s2[j]);
                }
        }
        if (s1[i] == '\0' && s2[j] == '\0')
        {
                return 0;
        }
        else if(s1[i] == '\0' || s2[i] == '\0')
        {
                return (s1[i] - s2[i]);
        }

}
```

```
void str_concat(char s1[], char s2[])
{
        int i,j;
        for(i=0;s1[i] != '\0';i++);

        for(j=0;s2[j] != '\0';i++,j++)
        {
                s1[i] = s2[j];
        }
        s1[i] = '\0';
}
```

```c
int btod(int bin)
{
  if(bin==0)
  {
    return 0;
  }
  else
  {
    return btod(bin/10)*2+(bin%10);
  }
}
int main()
{
  int binary,decimal;
  printf("enter the binary number");
  scanf("%d",&binary);
  decimal = btod(binary);
  printf("display the decimal value is : %d",decimal);
  return 0;
}
```

```
~/DelayedEmbellishedPort$ cc binary.c
binary.c: In function ‘main’:
binary.c:17:3: warning: ignoring return value of ‘s
ed with attribute ‘warn_unused_result’ [
   17 |   scanf("%d",&binary);
      |   ^~~~~~~~~~~~~~~~~~~
~/DelayedEmbellishedPort$ ./a.out
enter the binary number101
display the decimal value is : 5~/DelayedEmbellish
```

struct.c ×

```c
#include<stdio.h>
int main()
{
  struct student{
      int roll;
      char name[20];
      float marks_lang1;
      float marks_maths;
      float marks_comp;
      float avg;
  };
  int i,n;
  struct student class[20];
  printf("Enter the number of students in the class");
  scanf("%d",&n);
  printf("Enter the detail of student in the class\n");
  for(i=0;i<n;i++)
    {
      printf("Enter the %d student details :\n",i+1);
      printf("\n roll no :\n");
      scanf("%d",&class[i].roll);
      printf("\n name :\n");
      scanf("%s",class[i].name);
      printf("\n marks 1:\n");
      scanf("%f",&class[i].marks_lang1);
```

```
      |
struct.c:29:7: warning: ignoring return value of 'sc
ed with attribute 'warn_unused_result' [-Wunused-re
   29 |       scanf("%f",&class[i].marks_comp);
      |       ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~/DelayedEmbellishedPort$ ./a.out
Enter the number of students in the class2
Enter the detail of student in the class
Enter the 1 student details :

 roll no :
34

 name :
tasd

 marks 1:
56

 marks 2:
57

 marks 3:
58
Enter the 2 student details :

 roll no :
32

 name :
```

```
25      scanf("%f",&class[i].marks_lang1);
26      printf("\n marks 2:\n");
27      scanf("%f",&class[i].marks_maths);
28      printf("\n marks 3:\n");
29      scanf("%f",&class[i].marks_comp);
30     }
31     for(i=0;i<n;i++)
32       {
33         class[i].avg =
(class[i].marks_lang1+class[i].marks_maths+class[i].mark
s_comp)/3;
34       }
35       printf("List of the students greater than avg\n");
36       printf("roll no name avg.marks");
37       for(i=0;i<n;i++)
38         {
39           if(class[i].avg >= 35)
40             printf("%d%s%f\n",class[i].roll
41               ,class[i].name,class[i].avg);
42         }
43       printf("List of the students lesser thsn avg\n");
44       printf("roll no name avg.marks");
45       for(i=0;i<n;i++)
46         {
47           if(class[i].avg <= 25)
48             printf("%d%s%f\n",class[i].roll
```

```
 marks 1:
56

 marks 2:
57

 marks 3:
58
Enter the 2 student details

 roll no :
32

 name :
34

 marks 1:
35

 marks 2:
36

 marks 3:
37
List of the students greater than avg
roll no name avg.marks34tasd57.000000
323436.000000
List of the students lesser thsn avg
roll no name avg.marks~/DelayedEmbellishedPort
```

```
35      printf("List of the students greater than avg\n");
36      printf("roll no name avg.marks");
37      for(i=0;i<n;i++)
38        {
39          if(class[i].avg >= 35)
40            printf("%d%s%f\n",class[i].roll
41              ,class[i].name,class[i].avg);
42        }
43      printf("List of the students lesser thsn avg\n");
44      printf("roll no name avg.marks");
45      for(i=0;i<n;i++)
46        {
47          if(class[i].avg <= 25)
48            printf("%d%s%f\n",class[i].roll
49              ,class[i].name,class[i].avg);
50        }
51      return 0;
52  }
```

```c
#include<stdio.h>
int main()
{
    int a[10];
    int key,first,last,mid,n,j,i,temp;
    printf("Enter the number of elements:\n");
    scanf("%d",&n);
    printf("\n Enter the array of element:");
    for(i=0; i<n; i++)
    {
      scanf("%d",&a[i]);
    }
    for(i=0;i<n;i++)
    {
      for(j=0;j<n-1;j++)
      {
        if (a[j] > a[j+1])
        {
          temp = a[j];
          a[j] = a[j+1];
          a[j+1] = temp;
        }
      }
    }
    for(i=0;i<n;i++)
    {
```

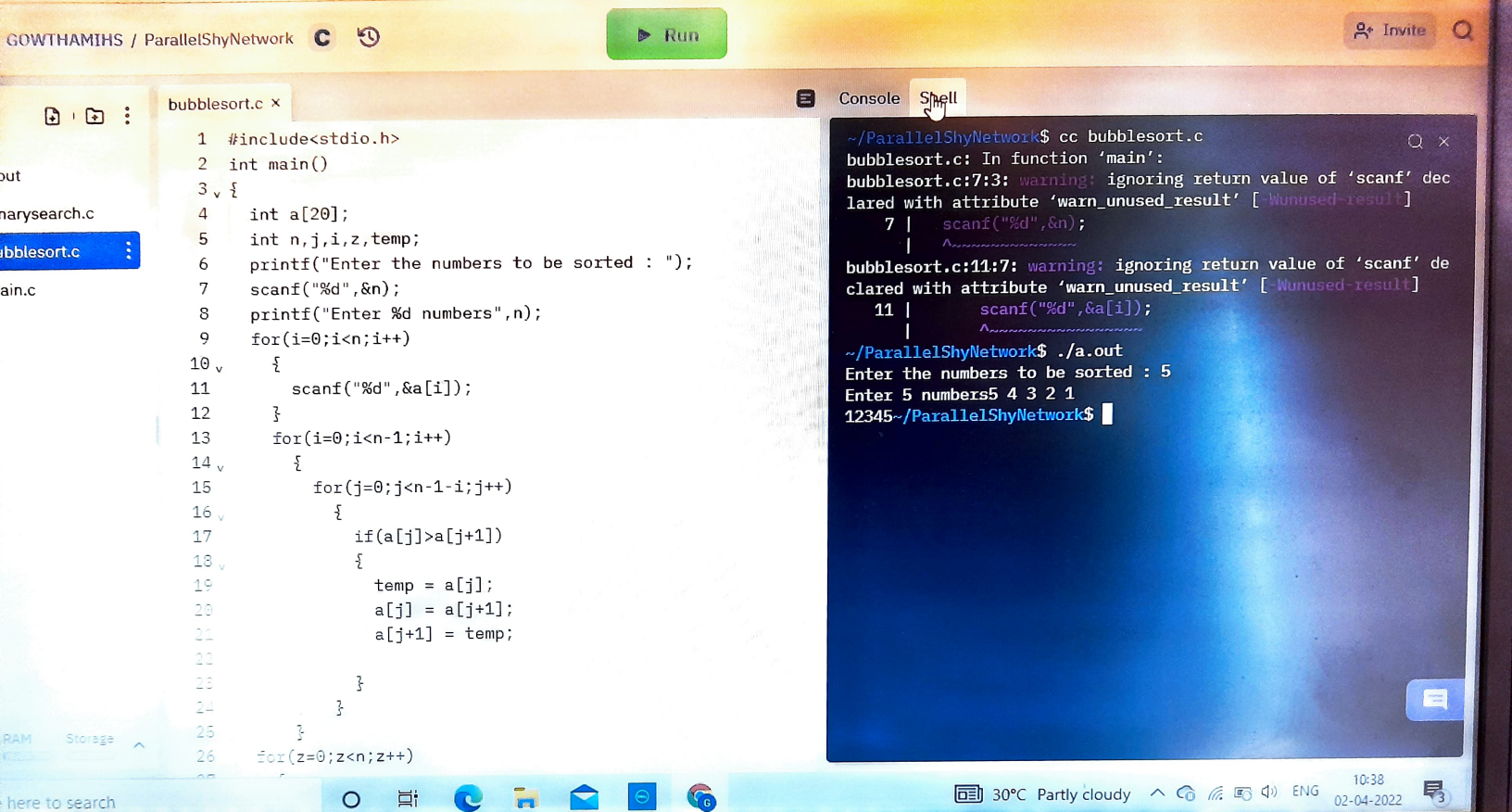
```c
int main() {
    int a[10];
    int key, first, last, mid, n, j, i, temp;
    printf("Enter the number of elements:\n");
    scanf("%d", &n);
    printf("\n Enter the array of element:");
    for (i = 0; i < n; i++) {
      scanf("%d", &a[i]);
    }
    for (i = 0; i < n; i++) {
      for (j = 0; j < n - 1; j++) {
        if (a[j] > a[j + 1]) {
          temp = a[j];
          a[j] = a[j + 1];
          a[j + 1] = temp;
        }
      }
    }
    for (i = 0; i < n; i++) {
      printf("\n The array of elements in sorted order %d",
a[i]);
    }
    printf("\n Enter the key element");
    scanf("%d", &key);
    first = 0;
```

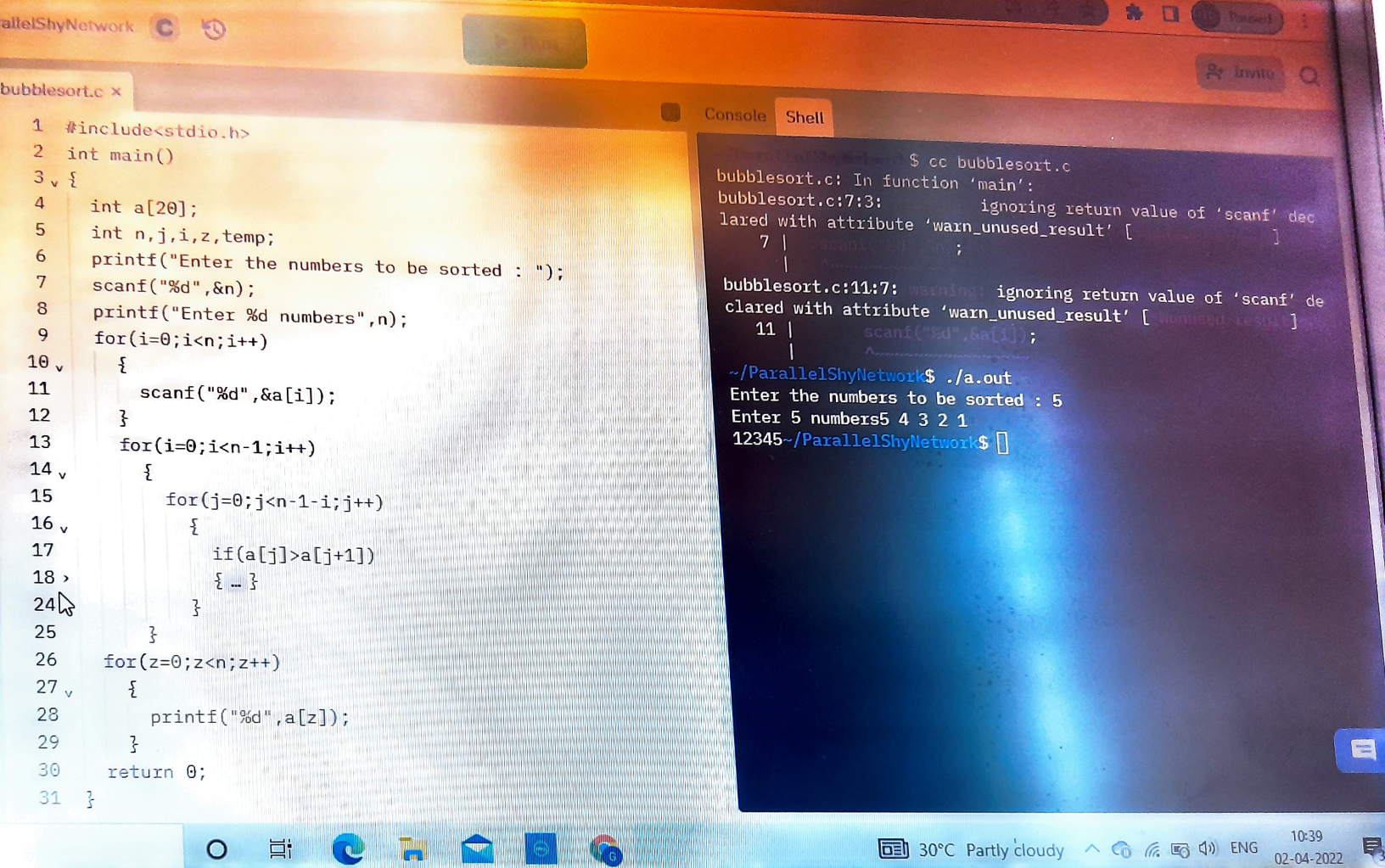
```c
#include <stdio.h>
#define METERCHARGE 100
#define PI 3.14156

int main() {
    char name[20];
    float units;
    float charge;
    float total;

    printf("Enter your name : \n");
    fgets(name, 20, stdin);
    printf("Enter units : \n");
    scanf("%f", &units);
    if (units < 0) {
        printf("INVALID INPUT\n");
        return 1;
    } else if (units <= 200) {
        charge = units * 0.80;
    } else if (units <= 300) {
        charge = 200 * 0.80 + (units - 200) * 0.90;
    } else
        (units > 300);
    { charge = 200 * 0.80 + 100 * 0.90 + (units - 300) * 1;
    }
    total = charge + METERCHARGE;
```

```c
#include<stdio.h>
int main()
{
    char opp;
    int num1,num2,result;
    printf("Enter the operator");
    scanf("%c",&opp);
    printf("Enter two integers");
    scanf("%d%d",&num1,&num2);
    if(opp=='+')
    {
      result=num1+num2;
    }
    else if(opp=='-')
    {
      result=num1-num2;
    }
    else if(opp=='*')
    {
      result=num1*num2;
    }
    else if(opp=='/')
    {
      if(num2==0)
      {
        printf("Divide by zero error\n");
```

main.c ×

```c
#include <stdio.h>
int main() {
    char opp;
    int num1, num2, result;
    printf("Enter the operator");
    scanf("%c", &opp);
    printf("Enter two integers");
    scanf("%d%d", &num1, &num2);
    if (opp == '+') {
        result = num1 + num2;
    } else if (opp == '-') {
        result = num1 - num2;
    } else if (opp == '*') {
        result = num1 * num2;
    } else if (opp == '/') {
        if (num2 == 0) {
            printf("Divide by zero error\n");
            return 2;
        }
        result = num1 / num2;
    } else if (opp == '%') {
        if (num2 == 0) {
            printf("Mode by zero error\n");
            return 3;
        }
        result = num1 % num2;
```


GOWTHAMIHS / ParallelShyNetwork
Run
Invite
bubblesort.c
Console
Shell
#include<stdio.h>
int main()
{
int a[20];
int n,j,i,z,temp;
printf("Enter the numbers to be sorted : ");
scanf("%d",&n);
printf("Enter %d numbers",n);
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
for(i=0;i<n-1;i++)
{
for(j=0;j<n-1-i;j++)
{
if(a[j]>a[j+1])
{
temp = a[j];
a[j] = a[j+1];
a[j+1] = temp;
}
}
}
for(z=0;z<n;z++)
~/ParallelShyNetwork$ cc bubblesort.c
bubblesort.c: In function 'main':
bubblesort.c:7:3: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
7 | scanf("%d",&n);
bubblesort.c:11:7: warning: ignoring return value of 'scanf' declared with attribute 'warn_unused_result' [-Wunused-result]
11 | scanf("%d",&a[i]);
~/ParallelShyNetwork$ ./a.out
Enter the numbers to be sorted : 5
Enter 5 numbers5 4 3 2 1
12345~/ParallelShyNetwork$
30°C Partly cloudy
ENG
10:38
02-04-2022



bubblesort.c

```c
#include<stdio.h>
int main()
{
  int a[20];
  int n,j,i,z,temp;
  printf("Enter the numbers to be sorted : ");
  scanf("%d",&n);
  printf("Enter %d numbers",n);
  for(i=0;i<n;i++)
    {
      scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
      {
        for(j=0;j<n-1-i;j++)
          {
            if(a[j]>a[j+1])
            { … }
          }
      }
   for(z=0;z<n;z++)
     {
       printf("%d",a[z]);
     }
   return 0;
}
```

Console Shell

```
$ cc bubblesort.c
bubblesort.c: In function 'main':
bubblesort.c:7:3:          ignoring return value of 'scanf' dec
lared with attribute 'warn_unused_result' [                    ]
    7 |                   ;
      |
bubblesort.c:11:7:         ignoring return value of 'scanf' de
clared with attribute 'warn_unused_result' [                   ]
   11 |       scanf("%d",&a[i]);
      |
~/ParallelShyNetwork$ ./a.out
Enter the numbers to be sorted : 5
Enter 5 numbers5 4 3 2 1
12345~/ParallelShyNetwork$
```

30°C Partly cloudy ENG 10:39 02-04-2022

bubblesort.c

```c
#include<stdio.h>
int main()
{
   int a[20];
   int n,j,i,z,temp;
   printf("Enter the numbers to be sorted : ");
   scanf("%d",&n);
   printf("Enter %d numbers",n);
   for(i=0;i<n;i++)
     {
       scanf("%d",&a[i]);
     }
     for(i=0;i<n-1;i++)
       {
         for(j=0;j<n-1-i;j++)
           {
             if(a[j]>a[j+1])
             { … }
           }
       }
   for(z=0;z<n;z++)
     {
       printf("%d",a[z]);
     }
   return 0;
}
```

Console Shell

```
                    $ cc bubblesort.c
bubblesort.c: In function 'main':
bubblesort.c:7:3:               ignoring return value of 'scanf' dec
lared with attribute 'warn_unused_result' [
    7 |                         ;
      |                                                          ]
bubblesort.c:11:7:              ignoring return value of 'scanf' de
clared with attribute 'warn_unused_result' [
   11 |       scanf("%d",&a[i]);
      |       ^~~~~~~~~~~~~~~~~                                  ]
~/ParallelShyNetwork$ ./a.out
Enter the numbers to be sorted : 5
Enter 5 numbers5 4 3 2 1
12345~/ParallelShyNetwork$
```

30°C Partly cloudy ENG 10:39 02-04-2022

matrix.c ×

```c
#include<stdio.h>
int main()
{
   int a[10][10],b[10][10],c[10][10],m,n,p,q,i,j,k;
   printf("enter the rows and column of A matrix\n");
   scanf("%d%d",&m,&n);
   printf("enter the rows and columns of B matrix\n");
   scanf("%d%d",&p,&q);
   if(n!=p)
   {
     printf("\n matrix miltiplication is not possible");
     return 0;
   }
   printf("enter the elementd of A matrix: \n");
   for(i=0;i<m;i++)
     {
       for(j=0;j<n;j++)
         {
           scanf("%d",&a[i][j]);
         }
     }
   printf("enter the elements of B matrix: \n");
   for(i=0;i<p;i++)
     {
       for(j=0;j<q;j++)
         {
```

```c
#include <stdio.h>
int main() {
  int a[10][10], b[10][10], c[10][10], m, n, p, q, i, j,
k;
  printf("enter the rows and column of A matrix\n");
  scanf("%d%d", &m, &n);
  printf("enter the rows and columns of B matrix\n");
  scanf("%d%d", &p, &q);
  if (n != p) {
    printf("\n matrix miltiplication is not possible");
    return 0;
  }
  printf("enter the elementd of A matrix: \n");
  for (i = 0; i < m; i++) {
    for (j = 0; j < n; j++) {
      scanf("%d", &a[i][j]);
    }
  }
  printf("enter the elements of B matrix: \n");
  for (i = 0; i < p; i++) { … }
  for (i = 0; i < m; i++) { … }
  printf("\n The resultant of matrix is :\n");
  for (i = 0; i < m; i++) { … }
  return 0;
}
```

```c
#include <stdio.h>
#include <math.h>
#define PI 3.142857
int main()
{
    float degree, x;
    float term, nume, deno;
    float sum;
    int i;
    printf("Enter the degree:\n");
    scanf("%f",&degree);
    x = degree * (PI / 180);
    sum = 0;
    nume = x;
    deno = 1;
    i = 1;

    do {
      term = nume / deno;
      sum = sum + term;
      i = i + 2;
      nume = -1 * nume * x * x;
      deno = deno * (i - 1) * i;
    } while (fabs(term) >= 0.00001);
    printf("The computed value of sin(%f)=%f\n", degree,
sum);
```

```c
#include <math.h>
#include <stdio.h>
#define PI 3.142857
int main() {
    float degree, x;
    float term, nume, deno;
    float sum;
    int i;
    printf("Enter the degree:\n");
    scanf("%f", &degree);
    x = degree * (PI / 180);
    sum = 0;
    nume = x;
    deno = 1;
    i = 1;

    do { … } while (fabs(term) >= 0.00001);
    printf("The computed value of sin(%f)=%f\n", degree,
sum);
    printf("Value from the library function is sin(%f)=
%f\n", degree, sin(x));
    return 0;
}
```