SOURCE CODE (BISECTION METHOD) -

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
float f(float x);
int main()
{
   float x1, x2, x0, e;
   printf("\n Enter the value of x1=");
   scanf("%f", &x1);
   printf("\n Enter the value of x2=");
   scanf("%f", &x2);
   printf("\n Enter the value of e=");
   scanf("%f", &e);
   if (f(x1) * f(x2) > 0)
   {
      printf("The guess values are incorrect\n");
      getch();
      return 1;
   }
   do
   {
      x0 = (x1 + x2) / 2;
      if (f(x1) * f(x0) < 0)
         x2 = x0;
      else
         x1 = x0;
   } while (fabs(f(x0)) >= e);
   printf("\n The root of the given equation is x0 = %f\n", x0);
   getch();
   return 0;
}
float f(float x)
{
   return (x * x - 25);
}
```

OUTPUT -

```
Enter the value of x1=4
Enter the value of x2=5
Enter the value of e=0.001
The root of the given equation is x0 = 4.999939
```

SOURCE CODE (NEWTON RAPHSON's METHOD) -

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define f(x) (x*x*x*x - x - 10)
#define d(x) (4*x*x*x - 1)
main()
{
  float x0,x1, e = 0.001,f1,d1;
printf("Enter the value of x0=") ;
scanf("%f", &x0);
f1= f(x0);
d1=d(x0);
x1=x0-(f1/d1);
while (fabs(f(x1))>e)
{
  x0=x1;
  f1=f(x0);
  x1=x0-(f1/d1);
}
printf("The root of given equation x1=\%f" ,x1);
getch();
}
```

OUTPUT -

```
Enter the value of x0=2
The root of given equation x1=1.855611
```

SOURCE CODE (TRAPEZOIDAL RULE) -

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
float f(float x) {
   return sqrt(1 - x * x);
}
void main() {
   float x0, xn, h, s1, s2 = 0, I, x;
   int i, n;
   printf("\n Enter the value of x0 = ");
   scanf("%f", &x0);
   printf("\n Enter the value of xn = ");
   scanf("%f", &xn);
   printf("\n Enter the value of n = ");
   scanf("%d", &n);
   h = (xn - x0) / n;
   s1 = f(x0) + f(xn);
   for (i = 1; i <= (n - 1); i++) {
      x = x0 + (i * h);
      s2 = s2 + f(x);
   }
   I = (h / 2) * (s1 + (2 * s2));
   printf("\n The integration of the given function is I = %f", I);
   getch();
}
```

OUTPUT -

```
 Enter the value of x0 = 0
 Enter the value of xn = 1
 Enter the value of n = 10
 The integration of the given function is I = 0.776130
```

SOURCE CODE (SIMPSON's 1/3rd RULE) -

```
#include <stdio.h>
#include <stdlib.h>
float f(float x) {
   return 11.25 * x * x;
}
int main() {
   float x0, xn, h, sum = 0, I;
   int i, n;
   printf("Enter the values of x0 and xn: ");
   scanf("%f %f", &x0, &xn);
   printf("Enter the value of n (must be even): ");
   scanf("%d", &n);
   if (n % 2 != 0) {
      printf("\nNumber of subdivisions should be even.\n");
      exit(0);
   }
   h = (xn - x0) / n;
   sum = f(x0) + f(xn);
   for (i = 1; i < n; i++) {
      float x = x0 + i * h;
      if (i % 2 == 0) {
         sum = sum + 2 * f(x);
      } else {
         sum = sum + 4 * f(x);
      }
   }
   I = (h / 3) * sum;
   printf("\nValue of the integration is I = %f\n", I);
   return 0;
}
```

OUTPUT -

```
 Enter the values of x0 and xn: 0 1
Enter the value of n (must be even): 100
Value of the integration is I = 3.750000
```

SOURCE CODE (FACTORIAL OF GIVEN NUMBER) -

```
#include <stdio.h>
factorial(int n) {
   if (n == 0) {
      return 1;
   } else {
      return n * factorial(n - 1);
   }
}
int main() {
   int number;
   printf("Enter a number: ");
   scanf("%d", &number);
   if (number < 0) {
      printf("Factorial is not defined for negative numbers.\n");
   } else {
      printf("Factorial of %d = %llu\n", number, factorial(number));
   }
   return 0;
}
```

OUTPUT -

```
Enter a number: 5
Factorial of 5 = 120
```

SOURCE CODE (PRESSURE BY BOYLE's LAW) -

```
#include <stdio.h>
int main() {
   float P, V, VF, R, T;
   R = 8.31;
   printf("\nEnter the temperature T: ");
   scanf("%f", &T);
   V = 0.1;
   VF = 1;
   printf("V        P\n");
   do {
      P = R * T / V;
      printf("%2.2f\t%4.2f\n", V, P);
      V = V + 0.1;
   } while (V <= VF);
   return 0;
}
```

OUTPUT -

```
Enter the temperature T: 300
V       P
0.10    24930.00
0.20    12465.00
0.30    8310.00
0.40    6232.50
0.50    4986.00
0.60    4155.00
0.70    3561.43
0.80    3116.25
0.90    2770.00
```