```
/*******************************************************************************
          Derivative using Newton's Divided Differences Table
*******************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 15

int main()
{
  int i, j, k, n;
  float x[MAX], f[MAX], a[MAX], d[MAX][MAX], xp, dif, sum, p;
  char q;
  printf("\n Input number of data points: ");
  scanf("%d", &n);
  printf("\n Input values of x and f(x) one set on each line: ");
  for(i=1;i<=n;i++)
    scanf("%f %f", &x[i], &f[i]);
  for(i=1;i<=n;i++)
    d[i][1]=f[i];
  for(j=2;j<=n;j++)
    for(i=1;i<=n-j+1;i++)
      d[i][j]=(d[i+1][j-1]-d[i][j-1])/(x[i+j-1]-x[i]);
  for(j=1;j<=n;j++)
    a[j]=d[1][j];
  do{
    printf("\n Input xp where derivative is required: ");
    scanf("%f", &xp);
    dif=a[2];
    for(k=3;k<=n;k++){
      sum=0.0;
      for(i=1;i<=k-1;i++){
        p=1.0;
        for(j=1;j<=k-1;j++){
          if(i!=j)
          p=p*(xp-x[j]);
        }
        sum=sum+p;
      }
      dif=dif+a[k]*sum;
    }
    printf("\n Derivative at x=%f is %f.", xp, dif);
    printf("\n Do you want to input another value? (y/n):");
    scanf(" %c", &q);
  }
  while (q=='y');
  getch();
  return 0;
}
```

```
/*******************************************************************************
                    First and Second derivatives using
                         Central Difference Formula
*******************************************************************************/

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

float f(float x){
  return x*x*exp(sqrt(x))*sin(x);
}

int main()
{
  float a, h, fd1, fd2;
  char q;
  do{
     printf("\nEnter the point at which derivatives are required: ");
     scanf("%f", &a);
     printf("\nEnter the value of h: ");
     scanf("%f", &h);
     fd1=(f(a+h)-f(a-h))/(2*h);
     fd2=(f(a+h)-2*f(a)+f(a-h))/(h*h);
     printf("\nThe first and second derivatives at x=%f are %f and %f respectively.", a, fd1, fd2);
     printf("\nDo you want to input another values?(y/n):");
     scanf(" %c", &q);
  }
  while (q=='y');
  getch();
  return 0;
}
```

```
/*************************************.....
```

```
/*******************************************************************************
        Integrating a Function using Composite Trapezoidal Rule
*******************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
[redacted]

float f(float x){
   return 1.0-exp(-x/2.0);
}

int main()
{
   int n, i;
   float a, b, h, sum=0.0, ict;
   printf("\n Give lower limit of integration a: ");
   scanf("%f", &a);
   printf("\n Give upper limit of integration b: ");
   scanf("%f", &b);
   printf("\n Give the segment width h:");
   scanf("%f", &h);
   n=(b-a)/h;
   for(i=1;i<=n-1;i++)
      sum=sum+2*f(a+i*h);
   sum=sum+f(a)+f(b);
   ict=sum*h/2;
   printf("\n Integration between %f and %f when h=%f is %f.", a, b, h, ict);
   getch();
   return 0;
}
```

```
/**********************************************************************************
        Integrating a Function using Composite Simpson's 1/3 Rule
**********************************************************************************/

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

float f(float x){
   return 1.0-exp(-x/2.0);
}

int main()
{
   int n, m, i;
   float a, b, h, sum=0.0, ics, x, [redacted];
   printf("\n Give lower limit of integration a: ");
   scanf("%f", &a);
   printf("\n Give upper limit of integration b: ");
   scanf("%f", &b);
   do{
      printf("\n Give number of segments n (Even number): ");
      scanf("%d", &n);
   }
   while (n%2!=0);
   h=(b-a)/n;
   m=n/2;
   for(i=1;i<=m;i++){
      x=a+(2*i-1)*h;
      sum=sum+4*f(x);
      if (i!=m)
      sum=sum+2*f(x+h);
   }
   sum=sum+f(a)+f(b);
   ics=sum*h/3.0;
   printf("\n Integration between %f and %f when h=%f is %f.", a, b, h, ics);
   getch();
   return 0;
}
```

```
/*****************************************************************************************
        Integrating a Function using Composite Simpson's 3/8 Rule
*****************************************************************************************/

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

float f(float x){
   return exp(-x*x);
}

int main()
{
   int n, m, i;
   float a, b, h, sum=0.0, ics, x;
   printf("\n Give lower limit of integration a: ");
   scanf("%f", &a);
   printf("\n Give upper limit of integration b: ");
   scanf("%f", &b);
   do{
      printf("\n Give number of segments n (divisible by 3): ");
      scanf("%d", &n);
   }
   while (n%3!=0);
   h=(b-a)/n;
   m=n/3;
   for(i=1;i<=m;i++){
      x=a+(3*i-2)*h;
      sum=sum+3*(f(x)+f(x+h));
      if (i!=m)
      sum=sum+2*f(x+2*h);
   }
   sum=sum+f(a)+f(b);
   ics=sum*3*h/8.0;
   printf("\n Integration between %f and %f when h=%f is %f.", a, b, h, ics);
   getch();
   return 0;
}
```