```
/*****************************************************************************
                    Approximation of root of nonlinear equation using
                              Fixed-Point Iteration Method
*****************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPS 0.00001

float g(float x){
   return 5+0.5*sin(x);
}

int main()
{
  float x0, x1;
  int n=1;
  printf("Enter an initial point x0:\n");
  scanf("%f", &x0);

  x1=g(x0);
  while (fabs((x1-x0)/x1)>EPS){
      n++;
      x0=x1;
      x1=g(x0);
      if (g(x1)==x1) break;
    }

   printf("The approximate root of given function is %f with %d number of iterations.", x1, n);
   getch();
   return 0;
}
```

```
/*****************************************************************************
                    Approximation of root of nonlinear equation using
                                Bisection Method
*****************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPS 0.00001

float f(float x){
   return x*x*x-2*sin(x);
}

int main()
{
   float a, b, c;
   int n=0;
   do{
      printf("Enter two valid initial points a and b:\n");
      scanf("%f %f", &a, &b);
    }
   while (f(a)*f(b)>=0);

   do{
      n++;
      c=(a+b)/2;
      if (f(c)==0) break;
      if (f(a)*f(c)<0) b=c;
      else a=c;
    }
   while (fabs((b-a)/a)>EPS);

   printf("The approximate root of given function is %f with %d number of iterations.", c, n);
   getch();
   return 0;
}
```

```
/*******************************************************************************
                    Approximation of root of nonlinear equation using
                                    Secant Method
*******************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPS 0.00001

float f(float x){
  return x*x*x-3*x+1;
}

int main()
{
   float x0, x1, x2;
   int n=0;
   do{
     printf("Enter two valid initial points x0 and x1:\n");
     scanf("%f %f", &x0, &x1);
    }
    while (f(x0)==f(x1));

    do{
      n++;
      x2=(x0*f(x1)-x1*f(x0))/(f(x1)-f(x0));
      if (f(x2)==0) break;
      x0=x1;
      x1=x2;
     }
     while (fabs((x1-x0)/x1)>EPS);

   printf("The approximate root of given function is %f with %d number of iterations.", x2, n);
   getch();
   return 0;
}
```

```
/*******************************************************************************
                    Approximation of root of nonlinear equation using
                              Newton-Raphson Method
*******************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPS 0.00001

float f(float x){
   return x*sin(x)+cos(x);
}

float fd(float x){
   return x*cos(x);
}

int main()
{
  float x0, x1;
  int n=1;
  do{
    printf("Enter a valid initial point x0:\n");
   scanf("%f", &x0);
   }
  while (fd(x0)==0);

   x1=x0-(f(x0)/fd(x0));
   while (fabs((x1-x0)/x1)>EPS){
       n++;
       x0=x1;
       x1=x0-(f(x0)/fd(x0));
       if (f(x1)==0) break;
      }

  printf("The approximate root of given function is %f with %d number of iterations.", x1, n);
  getch();
  return 0;
}
```

```
/*****************************************************************************
                    Approximation of root of polynomial using
                    Newton-Raphson Method with Horner's evaluation
*****************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPS 0.00001

void horner(int n, float a[], float x0, float p[]){
   int i;
   p[0]=a[n], p[1]=a[n];
   for (i=n-1; i>=1; i--){
      p[0]=a[i]+p[0]*x0;
      p[1]=p[0]+p[1]*x0;
   }
   p[0]=a[0]+p[0]*x0;
}

int main()
{
  float a[10], p[2], x0, x1;
  //p[0] and p[1] stores the value of the polynomial and its derivative at x0 respectively
  int i, n, count=1;
  printf("Enter the degree of the polynomial:\n");
  scanf("%d", &n);
  printf("\nEnter the coefficients of the polynomial starting from the highest degree:\n");
  for (i=n; i>=0; i--)
  scanf("%f", &a[i]);
  printf("Enter a valid initial point x0:\n");
  scanf("%f", &x0);

   horner(n, a, x0, p);
   x1=x0-(p[0]/p[1]);
   while (fabs((x1-x0)/x1)>EPS){
      count++;
      x0=x1;
      horner(n, a, x0, p);
      x1=x0-(p[0]/p[1]);
    }

  printf("\nThe approximate root of given function is %f with %d number of iterations.", x1, count);
  getch();
  return 0;
}
```