Date 1/9/22
Page No. 1

1. Bisection Method

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define f(x)  (x*x*x)+7*(x*x)+9
void main()
{   float x0, x1, x2, f0, f1, f2, e;
    int step = 1;
    printf("Enter two initial guesses");
    scanf("%f %f", &x0, &x1);
    printf("Enter the permissible Tolerance");
    scanf("%f", &e);
    f0 = f(x0);
    f1 = f(x1);
    if ((f0 * f1) > 0)
    {  printf("Incorrect Initial Guesses");
       printf("\n Press any key to exit");
        getch();
        exit(0);
    }
    printf("step\t\t x0 \t\t x1 \t\t x2");
    do
    {       x2 = (x0+x1)/2;
            f2 = f(x2);
           printf("\n%d\t\t%f\t\t%f\t\t%f",
                  step, x0, x1, x2);
          if((f0 * f2) < 0)
          {   x1 = x2;
              f1 = f2;
          }
```

```
        else
        {       x0 = x2 ;
                f0 = f2 ;
        }
        step++ ;
    } while ( fabs(f2) > e)
      printf(" \n The Root is %f", x2);
        getch();
}
```

Date 2 / 9 / 22
Page No. 3

2. Regula Falsi Method

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define f(x)    (x*x*x)-(2*x)-5
void main()
{    float x0, x1, x2, e, f0, f1, f2;
     int step = 1;
     printf("Enter two initial guesses");
     scanf("%f %f", &x0, &x1);
     printf("Enter the permissible Tolerance");
     scanf("%f", &e);
     f0 = f(x0);
     f1 = f(x1);
     if ((f0 * f1) > 0)
     {   printf("Incorrect initial guesses");
         return;
     }
     do
     {   x2 = (x0 * f1 - x1 * f0) / (f1 - f0);
         f2 = f(x2);
         printf("\n%d\t\t%f\t\t%f\t\t%f",
                step, x0, x1, x2);
         if ((f0 * f2) < 0)
         {    x1 = x2;
              f1 = f2;
         }
         else
         {    x0 = x2;
```

```
            f0 = f2;
        }
        step++;
    } while (fabs(f2) > e);
    printf("\nThe Root is %f", x2);
        getch();
}
```

Date 3/9/22
Page No. 5

3. Secant Method

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define f(x) (x*x*x) - (2*x) - 5
void main()
{   float x0, x1, x2, f0, f1, f2, e;
    int step = 1;
    printf("Enter two initial guesses");
    scanf("%f %f", &x0, &x1);
    printf("Enter the permissible Tolerance");
    scanf("%f", &e);
    f0 = f(x0);
    f1 = f(x1);
    if (f0 == f1)
    {   printf("Mathematical Error");
        return;
    }
    else {
    do
    {   x2 = (x0*f1 - x1*f0)/(f1-f0);
        printf("\n%d\t\t%f\t\t%f\t\t%f",
        step, x0, x1, x2);
        f2 = f(x2);
        x0 = x1;
        f0 = f1;
        x1 = x2;
        f1 = f2;
        step++;
```

```
    } while (fabs(f2) > e);
    printf("\n The Root is: %f", x2); }
    getch();
}
```

}

Date 4/9/22
Page No. 7

4. Newton Raphson Method

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define f(x)   (x*x*x)-(2*x)-5
#define vg(x)  (3*x*x)-2
void main()
{
    float x0, f0, vg0, x1, e;
    int step=1;
    printf("Enter one initial guess");
    scanf("%f", &x0);
    printf("Enter the Permissible Tolerance");
    scanf("%f", &e);
    f0 = f(x0);
    vg0 = vg(x0);
    if (vg0==0)
    {  printf("Mathematical Error");
       exit(0);
    }
    else {
     do
     {   x1 = x0-(f0/vg0);
         printf("\n%d\t\t%f\t\t%f", step, x0,
                x1);
         x0=x1;
         f0 = f(x0);
         vg0= vg(x0);
         step++;
     } while (fabs(f0)>e);
```

```
    printf("\n The Root is: %f", x1);
}
    getch();
}
```

5. Gauss Elimination Method

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{   float a[5][5], x[5], pivot, sum;
    int i, j, k, n;
    printf("Enter number of equations: ");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {   for(j=1; j<=n+1; j++)
        {   printf("Enter coeff. at %d %d
                position", i, j);
            scanf("%f", &a[i][j]);
        }
    }
    for(j=1; j<n; j++)
    {   for(i=j+1; i<n+1; i++)
        {   pivot = a[i][j]/a[j][j];
            for(k=1; k<=n+1; k++)
            {
                a[i][k] = a[i][k]-pivot* a[j][k];
            }
        }
    }
    printf("The upper triangular matrix is:\n");
    for(i=1; i<=n; i++)
    {   for(j=1; j<=n+1; j++)
        {   printf("%f\t", a[i][j]);
```

```
        }
        printf("\n");
    }
    for (i=n ; i>=1 ; i--)
    {
        sum = 0;
        for (j=i+1; j<=n ; j++)
        {
            sum = sum + a[i][j]* x[j];
        }
        x[i] = (a[i][n+1]-sum)/a[i][i];
    }
    printf(" The solution is \n");
        for (i=1 ; i<=n ; i++)
            printf("%f \n", x[i]);
            getch();
}
```

Date 6/9/22
Page No. 11

6. Gauss Jordan Method

```
# include <stdio.h>
# include <math.h>
# include <conio.h>
void main()
{   float a[5][5], x[5], pivot, sum;
    int i, j, k, n;
    printf("Enter number of equations: ");
    scanf("%d", &n);
    for (i=1; i<=n; i++)
    {       for (j=1; j<=n+1; j++)
            {
                printf("Enter coeff. at %d %d
                        position", i, j);
                scanf("%f", &a[i][j]);
            }
    }
  for( i=1; i<=n; i++)
  {     for(j=1; j<=n; j++)
        {   if(i!=j)
            {
                pivot = a[j][i]/a[i][i];
                for(k=1; k<=n+1; k++)
                {
                    a[j][k]= a[j][k]-pivot*a[i][k];
                }
            }
        }
  }
}
```

```
for (i=1 ; i<=n ; i++)
{
    x[i] = a[i][n+1] / a[i][i];
}
printf("The solution is \n");
for(i=1 ; i<=n ; i++)
  printf("%f" \n, x[i]);
  getch();
}
```

Date 7/9/22
Page No. 13

7. Gauss Seidel Method

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define f1(y,z)  (85-6*y+z)/27
#define f2(x,z)  (72-6*x-2*z)/15
#define f3(x,y)  (110-x-y)/54
void main()
{
    float x0, y0, z0, x1, y1, z1, e1, e2, e3, e;
    printf(" Enter the permissible error: ");
    scanf("%f", &e);
    x0 = 0;
    y0 = 0;
    z0 = 0;
    do
    {
        x1 = f1(y0, z0);
        y1 = f2(x1, z0);
        z1 = f3(x1, y1);
    printf("%f \t %f \t %f \n", x1, y1, z1);
    e1 = fabs(x1-x0);
    e2 = fabs(y1-y0);
    e3 = fabs(z1-z0);
    x0 = x1;
    y0 = y1;
    z0 = z1;
    } while (e1>e && e2>e && e3>e);
    getch();
}
```

Date 8 / 9 / 22
Page No. 14

8. Lagrange Interpolation

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{    float x[10], y[10], x1, total, total1=0;
     int i, j, n;
     printf("Enter number of items: ");
     scanf("%d", &n);
     for(i=1 ; i<=n ; i++)
     {   printf("Enter value of x: ");
         scanf("%f", &x[i]);
         printf("Enter the values of y: ");
         scanf("%f", &y[i]);
     }
   printf("Enter value of x for which to interpolate");
   scanf("%f", &x1);
   for(i=1 ; i<=n ; i++)
   {      total = 1;
        for(j=1 ; j<=n ; j++)
        {    if(i != j)
             {
                total = total *((x1-x[j])/(x[i]-x[j]));
             }
        }
     total1 = total1 + total * y[i] ;
   }
 printf("\n The answer is %f", total1);
   getch();
}
```

Date 9/9/22
Page No. 15

9. Newton's Forward Interpolation

```
#include <stdio.h>
#include <conio.h>
int fact (int n)
{    int f=1;
     while (n>0)
     {    f = f*n;
          n--;
     }  return f;
}
void main()
{    float a[10][10], x, y, h, p, px=1;
     int i, j, n;
     printf("Enter number of Data Items: ");
     scanf("%d", &n);
     printf("Enter the data: \n");
     for(i=0; i<n; i++)
     {    printf("X %d  ", i);
          scanf("%f", &a[i][0]);
          printf("Y %d  ", i);
          scanf("%f", &a[i][1]);
     }
     for(j=2; j<=n; j++)
     {    for(i=0; i<n-1; i++)
          {    a[i][j] = a[i+1][j-1] - a[i][j-1];
          }
     }
```

```
printf("The Difference Table is:\n");
for(i=0; i<=n-2; i++)
{   printf("\n");
    for(j=0; j<n+1-i; j++)
    {  printf("%.3f ", a[i][j]);
    }
}
printf("\n Enter the value for which you want
        to Interpolate: ");
scanf("%f", &x);
 h = a[1][0] - a[0][0];
 y = a[0][1];
 p = (x - a[0][0]) / h;
 for(i=1; i<n; i++)
 {   px = px * (p-(i-1));
     y = y + (a[0][i+1] * px) / fact(i);
 }
 printf("Answer is %.3f", y);
}
```

Date 10/9/22
Page No. 17

10. Newton's Backward Interpolation

```
#include <stdio.h>
#include <conio.h>
void main()
{   int fact(int n)
    {   int f=1;
        while(n>0)
        {   f = f*n;
            n--;
        } return f;
    }
    float a[10][10], x, y, p, h, px=1;
    int i, j, n;
    printf("Enter number of Data Items: ");
    scanf("%d", &n);
    printf("Enter the data:\n");
    for(i=0; i<n; i++)
    {   printf("X %d  ", i);
        scanf("%f", &a[i][0]);
        printf("Y %d  ", i);
        scanf("%f", &a[i][1]);
    }
    for(j=2; j<=n; j++)
    {   for(i=0; i<n-1; i++)
        {   a[i][j] = a[i+1][j-1] - a[i][j-1];
        }
    }
```

```
printf("The Difference Table is:\n");
 for(i=0 ; i<=n-2 ; i++)
 {    printf("\n");
      for(j=0 ; j<n+1-i ; j++)
      {  printf("%f\t", a[i][j]);
      }
 }
printf("\n Enter the value of x for which
        you want to interpolate: ");
scanf("%f", &x);
h = a[n-1][0] - a[n-2][0];
y = a[n-1][1];
p = (x- a[n-1][0]) / h;
for(i=1 ; i<n ; i++)
{     px = px * (p+(i-1));
      y = y + (a[n-1-i][i+1]*px) / fact(i);
}
printf("\n The result after Interpolation is %f",y);
     getch();
}
```

Date 11/09/22
Page No. 19

11. Trapezoidal Rule

```
#include <stdio.h>
float f(float x)
{   return (1.0/(1+x*x));
}
void main()
{   int i,n;
    float a, b, h, sum, vans;
    printf("Enter the limits: ");
    scanf("%f %f", &a, &b);
    printf("Enter number of sub-intervals: ");
    scanf("%d", &n);
    sum = f(a) + f(b);   h = (b-a)/n;
    for(i=1; i<=n-1; i++)
    {   sum = sum + 2*f(a+i*h);
    }
    vans = (sum*h)/2;
    printf("Answer is %f ", vans);
    getch();
}
```

Date 12/09/22
Page No. 20

12. Simpson's 1/3 Rule

```
#include <stdio.h>
float f(float x)
{ return (1.0/(1+x*x));
}
void main()
{   int i, n;
    float a, b, h, sum, ans;
    printf("Enter the limits: ");
    scanf("%f %f", &a, &b);
    printf("Enter number of sub-intervals: ");
    scanf("%d", &n);
    h = (b-a)/n;
    sum = f(a) + f(b);
    for(i=1; i<=n-1; i=i+2)
    {   sum = sum + 4*f(a+i*h);
    }
    for(i=2; i<=n-2; i=i+2)
    {   sum = sum + 2*f(a+i*h);
    }
    ans = (sum*h)/3;
    printf("Answer is %f", ans);
     getch();
}
```

13. Simpson's 3/8 Rule

```
#include <stdio.h>
float f(float x)
{   return (1.0/(1+x*x));
}
void main()
{   int i,n;
    float a, b, h, sum, ans;
    printf("Enter the limits: ");
    scanf("%f %f", &a, &b);
    printf("Enter number of sub-intervals: ");
    scanf("%d", &n);
    h = (b-a)/n;
    sum = f(a) + f(b);
    for(i=1; i<=n-1; i=i+3)
    {   sum = sum + 3*f(a+i*h)+3*f(a+(i+1)*h);
    }
    for(i=3; i<=n-3; i=i+3)
    {   sum = sum + 2*f(a+i*h);
    }
    ans = (3*sum*h)/8;
    printf("Answer is %f", ans);
    getch();
}
```

Date 14/09/22
Page No. 22

14. Weddle's Rule

```
#include <stdio.h>
float f(float x)
{
    return (1.0/(1+x*x));
}
void main()
{
    int i, n;
    float a, b, h, sum, ans;
    printf("Enter the limits: ");
    scanf("%f %f", &a, &b);
    printf("Enter the number of sub-intervals: ");
    scanf("%d", &n);
    h = (b-a)/n;
    sum = 0;
    for(i=0; i<=n; i=i+6)
    {
    sum = sum + f(a+i*h) + 5*f(a+(i+1)*h)+
    f(a+(i+2)*h) + 6*f(a+(i+3)*h) + f(a+(i+4)*h)
    + 5*f(a+(i+5)*h) + f(a+(i+6)*h);

    }
    ans = (3*sum*h)/10;
    printf("Answer is %f", ans);
    getch();
}
```