**SIMPSON’S 1/3RD METHOD**

**EXAMPLE 1: Evaluate F(x)=, n=6**

**PROGRAM:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float y[50],a,b,x,h,s1,s2,sum1,simp1;**

**int i,n;**

**clrscr();**

**fin=fopen("sim13.in","r");**

**fout=fopen("sim13.out","w");**

**fscanf(fin,"%f%f%d",&a,&b,&n);**

**h=(b-a)/n;**

**x=a;**

**for(i=1;i<=n+1;i++)**

**{**

**y[i]=(sin(x)-log(x)+exp(x));**

**fprintf(fout,"\n value of y[%.2f] is %f ",x,y[i]);**

**printf("\n value of %f is %f ",x,y[i]);**

**x=x+h;**

**}**

**s1=0;**

**for(i=2;i<=n;i=i+2)**

**{**

**s1=s1+y[i];**

**}**

**s2=0;**

**for(i=3;i<=n;i=i+2)**

**{**

**s2=s2+y[i];**

**}**

**sum1=(y[1]+(4\*s1)+(2\*s2)+y[n+1]);**

**simp1=(h/3)\*(sum1);**

**fprintf(fout,"\n the integration of f(x) %f ",simp1);**

**getch();**

**}**

**INPUT FILE**

**0.2**

**1.4**

**6.0**

**output file**

**value of y[0.20] is 3.029510**

**value of y[0.40] is 2.797534**

**value of y[0.60] is 2.897587**

**value of y[0.80] is 3.166040**

**value of y[1.00] is 3.559753**

**value of y[1.20] is 4.069834**

**value of y[1.40] is 4.704177**

**the integration of f(x) 4.052133**

**EXAMPLE 2: Evaluate F(x)= , n=5**

**PROGRAM**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float y[50],a,b,x,h,s1,s2,sum1,simp1;**

**int i,n;**

**clrscr();**

**fin=fopen("s13a.in","r");**

**fout=fopen("s13a.out","w");**

**fscanf(fin,"%f%f%d",&a,&b,&n);**

**h=(b-a)/n;**

**x=a;**

**for(i=1;i<=n+1;i++)**

**{**

**y[i]=(exp(x)+exp(-x))/2.0;**

**fprintf(fout,"\n value of y[%.2f] is %f ",x,y[i]);**

**printf("\n value of %f is %f ",x,y[i]);**

**x=x+h;**

**}**

**s1=0;**

**for(i=2;i<=n;i=i+2)**

**{**

**s1=s1+y[i];**

**}**

**s2=0;**

**for(i=3;i<=n;i=i+2)**

**{**

**s2=s2+y[i];**

**}**

**sum1=(y[1]+(4\*s1)+(2\*s2)+y[n+1]);**

**simp1=(h/3)\*(sum1);**

**fprintf(fout,"\n the integration of f(x) %f ",simp1);**

**getch();**

**}**

**INPUT**

**0.0**

**1.0**

**5.0**

**OUTPUT**

**value of y[0.00] is 1.000000**

**value of y[0.20] is 1.020067**

**value of y[0.40] is 1.081072**

**value of y[0.60] is 1.185465**

**value of y[0.80] is 1.337435**

**value of y[1.00] is 1.543081**

**the integration of f(x) 1.080148**

**EXAMPLE 3: Evaluate taking h=0.5 find the inherent error in**

**your calculation and also obtain minimum number of sub**

**intervals , so that value of integral will be correct up to 5**

**places of decimal**

**PROGRAM**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float y[50],a,b,x,h,s1,s2,sum1,simp1;**

**int i,n;**

**clrscr();**

**fin=fopen("s13b.in","r");**

**fout=fopen("s13b.out","w");**

**fscanf(fin,"%f%f%d",&a,&b,&n);**

**h=(b-a)/n;**

**x=a;**

**for(i=1;i<=n+1;i++)**

**{**

**y[i]=(log10(x));**

**fprintf(fout,"\n value of y[%.2f] is %f ",x,y[i]);**

**printf("\n value of %f is %f ",x,y[i]);**

**x=x+h;**

**}**

**s1=0;**

**for(i=2;i<=n;i=i+2)**

**{**

**s1=s1+y[i];**

**}**

**s2=0;**

**for(i=3;i<=n;i=i+2)**

**{**

**s2=s2+y[i];**

**}**

**sum1=(y[1]+(4\*s1)+(2\*s2)+y[n+1]);**

**simp1=(h/3)\*(sum1);**

**fprintf(fout,"\n the integration of f(x) %f ",simp1);**

**getch();**

**}**

**INPUT**

**1.0**

**5.0**

**8.0**

**OUTPUT**

**value of y[1.00] is 0.000000**

**value of y[1.50] is 0.176091**

**value of y[2.00] is 0.301030**

**value of y[2.50] is 0.397940**

**value of y[3.00] is 0.477121**

**value of y[3.50] is 0.544068**

**value of y[4.00] is 0.602060**

**value of y[4.50] is 0.653212**

**value of y[5.00] is 0.698970**

**the integration of f(x) 1.757440**

**EXAMPLE 4: Evaluate F(x)= dx , n=5**

**PROGRAM**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float y[10],a,b,x,h,s1,s2,sum1,simp1;**

**int i,n;**

**clrscr();**

**fin=fopen("f1.in","r");**

**fout=fopen("f1.out","w");**

**fscanf(fin,"%f %f %d ",&a,&b,&n);**

**h=(b-a)/n;**

**/\* loop for x and y(i) value \*/**

**x=a;**

**for(i=1;i<=n+1;i++)**

**{**

**/\* write your function \*/**

**y[i]=(sin(x)/sqrt(x));**

**if(x==0.0)**

**{**

**y[i]=0.0;**

**}**

**fprintf(fout,"\n value of y[%.2f] is %f ",x,y[i]);**

**x=x+h;**

**}**

**s1=0;**

**/\* loop for odd value of y(i) addition \*/**

**for(i=2;i<=n;i=i+2)**

**{**

**s1=s1+y[i];**

**}**

**s2=0;**

**/\* loop for even value of y(i) addition \*/**

**for(i=3;i<=n;i=i+2)**

**{**

**s2=s2+y[i];**

**}**

**/\* simpsion formulation \*/**

**sum1=(y[1]+(4\*s1)+(2\*s2)+y[n+1]);**

**simp1=(h/3)\*(sum1);**

**fprintf(fout,"\n The integration of y(x) %f",simp1);**

**getch();**

**}**

**INPUT**

**0.0**

**1.0**

**5**

**OUTPUT**

**value of y[0.00] is 0.000000**

**value of y[0.20] is 0.444238**

**value of y[0.40] is 0.615724**

**value of y[0.60] is 0.728950**

**value of y[0.80] is 0.802028**

**value of y[1.00] is 0.841471**

**The integration of y(x) 0.557982**

**EXAMPLE 5:**

**Calculate specific heat of cu at t=300 K,given that θ=340K,N=12 also find inherent error in computation by taking half as many no of sub intervals**

**Cv=9Rdθ**

**PROGRAM**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float y[20],a,b,x,h,s1,s2,sum,th,t,r,cont,cv,simp;**

**int i,n;**

**clrscr();**

**fin=fopen("sh.in","r");**

**fout=fopen("sh.out","w");**

**fscanf(fin,"%f %f %d %f %f %f",&a,&b,&n,&th,&t,&r);**

**h=(b-a)/n;**

**/\* loop for x and y(i) value \*/**

**x=a;**

**for(i=1;i<=n+1;i++)**

**{**

**y[i]=((x\*x\*x\*x\*exp(x))/((exp(x)-1.0)\*(exp(x)-1.0)));**

**if(x==0.0)**

**{**

**y[i]=0.0;**

**}**

**fprintf(fout,"\n value of y[%.2f] is %f ",x,y[i]);**

**x=x+h;**

**}**

**s1=0;**

**/\* loop for odd value of y(i) addition \*/**

**for(i=2;i<=n;i=i+2)**

**{**

**s1=s1+y[i];**

**}**

**s2=0;**

**/\* loop for even value of y(i) addition \*/**

**for(i=3;i<=n-1;i=i+2)**

**{**

**s2=s2+y[i];**

**}**

**/\* simpsion formulation \*/**

**sum=(y[1]+(4\*s1)+(2\*s2)+y[n+1]);**

**simp=(h/3)\*(sum);**

**cont=9\*r\*(pow((t/th),3));**

**cv=cont\*simp;**

**fprintf(fout,"\n const= %f & simp1=%f",cont,simp);**

**fprintf(fout,"\n the specific heat of cu at t=300 k %f ",cv);**

**getch();**

**}**

**INPUT**

**0**

**1.1333**

**12**

**340**

**300**

**8.314**

**OUTPUT**

**value of y[0.00] is 0.000000**

**value of y[0.09] is 0.008913**

**value of y[0.19] is 0.035571**

**value of y[0.28] is 0.079738**

**value of y[0.38] is 0.141023**

**value of y[0.47] is 0.218883**

**value of y[0.57] is 0.312637**

**value of y[0.66] is 0.421467**

**value of y[0.76] is 0.544434**

**value of y[0.85] is 0.680489**

**value of y[0.94] is 0.828484**

**value of y[1.04] is 0.987187**

**value of y[1.13] is 1.155299**

**const= 51.401943 & simp1=0.455407**

**the specific heat of cu at t=300 k 23.408819**

**EXAMPLE 6:**

**A missile launched from a ground station. The acceleration during its 1st 80 sec of light as recorded .Compute the velocity of missile.**

**PROGRAM:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float y[10],a,b,x[10],h,s1,s2,sum1,simp1;**

**int i,n;**

**clrscr();**

**fin=fopen("i2.in","r");**

**fout=fopen("i2.out","w");**

**fscanf(fin,"%f %f %d",&a,&b,&n);**

**h=(b-a)/n;**

**/\* loop for read the value x(i) & y(i) \*/**

**for(i=1;i<=n+1;i++)**

**{**

**fscanf(fin,"\n %f\t %f ",&x[i],&y[i]);**

**}**

**s1=0;**

**/\* loop for odd value of y(i) addition \*/**

**for(i=2;i<=n;i=i+2)**

**{**

**s1=s1+y[i];**

**}**

**s2=0;**

**/\* loop for even value of y(i) addition \*/**

**for(i=3;i<=n-1;i=i+2)**

**{**

**s2=s2+y[i];**

**}**

**/\* simpsion formulation \*/**

**sum1=(y[1]+(4\*s1)+(2\*s2)+y[n+1]);**

**simp1=(h/3)\*(sum1);**

**fprintf(fout,"\n the integration of f(x) %f",simp1);**

**getch();**

**}**

**INPUT**

**0.0**

**80.0**

**8.0**

**0 30**

**10 31.63**

**20 33.34**

**30 35.47**

**40 37.75**

**50 40.33**

**60 43.25**

**70 46.69**

**80 50.67**

**OUTPUT**

**the integration of f(x) 3200.000000**