

PROGRAM STATEMENT: Evaluate $\int_0^1 (1/\sqrt{1+x^2})$ taking 10 intervals by Trapezoidal Rule

THEORY: It is two points quadrature, i.e., $n=1$ (n is the number of intervals). Therefore there are only two functional values $y_0=f(x_0)=f(a)$, $y_1=f(x_n)=f(b)$, where $b-a=h$ (where h is the difference between two intervals). Thus the second and higher order differences are not possible. Substituting $n=1$ and neglecting the second and higher order differences we get, $I_T = h/2 * (y_0 + y_1)$. This formula is known as Trapezoidal rule for integration. The composite Simpson's One-third Rule for numerical integration is...

$$I_C = h/2 * [\text{Sum of first and last ordinates} + 2(\text{Sum of all other ordinates})]$$

PROGRAM CODE:

```
//C Program to implement Trapezoidal formula for Numerical Integration
#include <stdio.h>
#include <math.h>
#define EQ (1/(sqrt(1+pow(x,2))))
int main()
{
    int i=0,n;
    double x,t=0,h,u,l;
    printf("The equation we are solving is-> y=1/(sqrt(1+x^2))\n");
    printf("\n\tEnter upper limit::");
    scanf("%lf",&u);
    printf("\n\tEnter lower limit::");
    scanf("%lf",&l);
    printf("\n\tEnter number of intervals::");
    scanf("%d",&n);
    h=(u-l)/n;
    x=l;
    double a[n+1];
    for(i=0;i<=n;i++)
    {
        a[i]=EQ;
        printf("\n\t%lf\t%lf",x,a[i]);
        x=x+h;
        if((i==0)||(i==n))
        {
            t=t+a[i];
        }
        else
        {
            t=t+(2*a[i]);
        }
    }
    t=t*h/2;
    printf("\n\tValue by Trapezoidal rule is %lf ",t);
    return 0;
}
```

OUTPUT:

The equation we are solving is->y=1/(sqrt(1+x^2))

Enter upper limit::1

Enter lower limit::0

Enter number of intervals::10

0.000000	1.000000
0.100000	0.995037
0.200000	0.980581
0.300000	0.957826
0.400000	0.928477
0.500000	0.894427
0.600000	0.857493
0.700000	0.819232
0.800000	0.780869
0.900000	0.743294
1.000000	0.707107

Value by Trapezoidal rule is 0.881079