5) (c) write a program to implement trapezoidal rule to evaluate $\int_{0}^{10} (1 - e^{-x/2}) dx$

⟹
```c
#include <stdio.h>
#include <conio.h>
#include <Math.h>
void main()
{
    float a,b,h,x,y,y0,yn,xn,s,r;
    int i,n;
    float f(float);
    clrscr();
    printf("\n\n Enter the lower limit: ");
    scanf("%f",&a);
    printf("\n\n Enter the upper limit: ");
    scanf("%f",&b);
    printf("\n\n Enter the interval: ");
    scanf("%d",&n);
    h = (b-a)/n;
    y0 = f(a);
    yn = f(b);
    x = a+h;
    s = 0;
    for(i=1; i<=(n-1); i++)
    {
        y = f(x);
        s = s+y;
        x = x+h;
    }
    r = (h/2) * ((y0+yn)+(2*s));
    printf("\n\n The result is: %f", r);
    getch();
}
float f(float x)
{
    return (1 - exp(-x/2))
}
```

7) (C) Draw a flow chart for interpolation using Newton's forward difference formula.

⇒ Flow chart ⇒

(start)

↓

/ input n /

↓

/ Get values of ax, ay /

↓

(Get values of x)

↓

[h = ax[1] - ax[0]]

↓

[Start loop i=0 to (n-1)]

↓

[diff.[i][1] = ay[i+1] - ay[i]]

↓

(End loop i)

↓

[Start loop j=2 to 4]

↓

/ start loop i=0 to n-j /

↓

[diff[i][j] = diff[i+1][j-1] - diff[i][j-1]]

↓

(end loop i)

↓

(end loop j) → (i=0)

↓

(A) → [is ax[i] > x] —NO→ [i = i+1] → (A)

↓ yes

[i = i-1] → [p = (x - ax[i])/h]

↓

[y₁ = p * diff[i-1][1]]

[y₂ = p*(p+1)*diff[i-1][2]/2] ←

↓

[y₃ = (p+1)*p*(p-1)*diff[i-2][3]/6]

↓

[y₄ = (p+2)*(p+1)*p*(p-1)*diff[i-3][4]/24]

↓

[y = ay[i] + y₁+ y₂ + y₃ + y₄] → [print output x, y]

↓

(stop)