```
/*
Question:

1. Solve the heat diffusion equation for a rectangular plate
 with following boundary conditions:

(i) along the left edge the temperature is given,
    it is time independent,and changes linearly along the edge

(ii) below a constant heat flux is given

(iii) along the other two edges there is a perfect heat insulation.
      The initial condition for temperature: linear in both coordinates
      and satisfy boundary condition along left edge.

mathematical description:

The heat diffusion equation is ......
δT/δt = a^2*(δ2T/δx2 + δ2T/δy2)

where, T(x,y,t) is temperature function .
 and a^2 = k/sϱ is the diffusivity of a substance,
k= coefficient of conductivity of material,
ϱ= density of the material, and
s= specific heat capacity.

(δ2T/δx2 + δ2T/δy2)= δT/δt

boundary conditions ....(i) T(0,y,t)=T1*y
                        (ii)T(x,0,t)=T2
                        (iii)T(x,y,0)=T3*x+T4*y
where T1,T2,T3,T4 are constants.


δ2T/δx2 + δ2T/δy2=0.....(T is independent of time along left to right.it becomes
Laplace equation)


numerical solution:

we shall assume at first that the plate is so long compared to its width that
we may make the mathematical approximation that
it extends to infinity in the y direction. It is then
called a semi-infinite plate. This is a good approximation if we are interested in
temperatures not too
near the far end.
The temperature T satisfies Laplace's equation
inside the plate where there are no sources of heat,
that is,

∇2T = 0,
 OR, δ2T/ δx2  + δ2T/ δy2 = 0

The program below for Solution of Laplace equation in C language is based
on the finite difference approximations to derivatives in which the
xy-plane is divided into a network of rectangular of sides Δx=h
and Δy=k by drawing a set of lines.

x=ih, i=0,1,2,…
y=jk, j=0, 1,  2….

The points of intersection of these families of lines are called mesh points,
lattice points or grid points.


*/

#include<stdio.h>
#include<math.h>
#define S 4

typedef float new[ +1][ +1];
void       (int  ,new  )
{
    int  ;
          ("\n Enter the value of T[%d,j],i=1,%d\n", , );
    for( =1; <= ; ++)
         ("%f",& [ ][ ]);
}
void       (int  , new  )
{
int  ;
      ("Enter the value of T[i,%d],""i=2,%d\n", , -1);
for( =2; <= -1; ++)
```

```
      ("%f",& [ ][ ]);
}
void     (new   , int    , int     )
{
    int   , ;
    for( = ; <= ; ++)
    {
        for( =1; <= ; ++)
              ("%d,%d,%f",  ,     ,   [ ][ ]);
              ("\n");

    }
}
     ()
{
    new  ;
    float    ,   ,   ,   ;
   int   , ,    ,        ;
   for( =1; <= ; ++)
   for( =1; <= ; ++)
    [ ][ ]=0;
         ("\n  Enter the Boundary Condition\n");
         (1, );       ( , );
         (1, );       ( , );
         (" Enter the allowed error and maximum number of iteration : ");
        ("%f%f",&   ,&      );
   for(   =1;   <=      ;   ++)
   {
          =0;
       for( =2; < -1; ++)
       {
       for( =2; <= -1; ++)
       {
            =( [ -1][ ]+ [ +1][ ]+ [ ][ +1]+ [ ][ -1])/4;
            =    ( [ ][ ]- );
           if( >   )
              = ;
            [ ][ ]= ;
       }
             (" Iteration Number %d\n",   );
           ( ,9,2);
       if(   <=  )
       {
                 (" After %d iteration \n The solution : \n",   );
               ( ,8,1);
           return 0;

       }
   }
         (" Sorry! The number of iteration is not sufficient");
   return 1;
}
}

/*the constants T1,T2,T3,T4 are chosen as we want here.

OUTPUT:

Enter the Boundary Condition

Enter the value of T[1,j],j=1,4
10
10
10
10

 Enter the value of T[4,j],j=1,4
10
5
0
0
Enter the value of T[j,1],j=2,3
20
20
Enter the value of T[j,4],j=2,3
5
0
 Enter the allowed error and maximum number of iteration : 0.1
1
 Iteration Number 1

 Sorry! The number of iteration is not sufficient
```

```
167
168
169   */
170
```