#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
#include <sys/time.h>  
#include "time.h"  
int main(int argc, char \*argv[])  
{  
//--------------------Declaring Variables-------------------------  
 int max = 1024, i, j , lp;  
 int top, bottom, left, right;  
 float net[1024][1024];  
 float x, y, fL, fR, fB, fT;  
 unsigned int par=160000, loop=2000;  
 struct timespec start, stop;  
 double t1=0, t2=0, result=0;  
//----------------------------------------------------------------  
//------------------calculate Starting time-----------------------  
 clock\_gettime(CLOCK\_REALTIME,&start);  
 t1 = start. tv\_sec + (start. tv\_nsec/pow(10, 9));  
//----------------------------------------------------------------  
//------------------Initialising grid-----------------------------  
 for (i=0; i<max; i++)  
 for (j =0; j <max; j ++)  
 net[i][j ]=0;  
//----------------------------------------------------------------  
//------------------ Mapping--------------------------------------  
 for (lp=1; lp<loop; lp++){  
 for ( i = 0; i < par; ++i){  
 //---Random position to particle---  
 x = ((float) rand()/(float)(RAND\_MAX) \* (float) max);  
 y = ((float) rand()/(float)(RAND\_MAX) \* (float) max);  
 //\_\_\_finding coordinate around particle\_\_\_  
 left = (int) floor(x);  
 right = left + 1;  
 bottom = (int) floor(y);  
 top = bottom +1;  
 //\_\_\_Checking boundary conditions\_\_\_  
 if (top>=max|| bottom>=max|| left>=max|| right>=max)  
 continue;  
 //\_\_\_Finding particle position within box\_\_\_  
 fL = x - left;  
 fR = 1 - fL;  
 fB = y - bottom;  
 fT = 1 - fB;  
 //\_\_\_calculating contribution\_\_\_  
 net[left][bottom] = net[left][bottom] +( fT \* fR ) ;  
 net[right][bottom] = net[right][bottom] +( fT \* fL ) ;  
 net[left][top] = net[left][top] +( fB \* fR ) ;  
 net[right][top] = net[right][top] +( fB \* fL ) ;  
 }  
 }  
//----------------------------------------------------------------  
//------------------calculate End time----------------------------  
 clock\_gettime(CLOCK\_REALTIME,&stop);  
 t2 = stop. tv\_sec + (stop. tv\_nsec/pow(10, 9));  
//----------------------------------------------------------------  
//----------------calculating processing time---------------------  
 result = t2 - t1 ;  
 printf("its done in :\t%lf s\n", result);  
//----------------------------------------------------------------  
//--------------- Saving result in file---------------------------  
//\_\_\_Opening file\_\_\_  
 FILE \*f = fopen("file1.txt", "w");  
 par\*=loop;  
 if (f == NULL){  
 printf("Error opening file!\n");  
 exit(1);  
 }  
//\_\_\_Normalizing result\_\_\_  
 float avg= par/(max\*max);  
 for ( i = 0; i < max; ++i){  
 for ( j = 0; j < max; j ++){  
 fprintf (f, "%f ,",((net[i][j ])/avg));  
 }  
 fprintf (f, "\n" );  
 }  
//\_\_\_Closing file\_\_\_  
 fclose(f);  
//------------------------------------------------------------  
return 0;  
}