

```

#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;
}
```