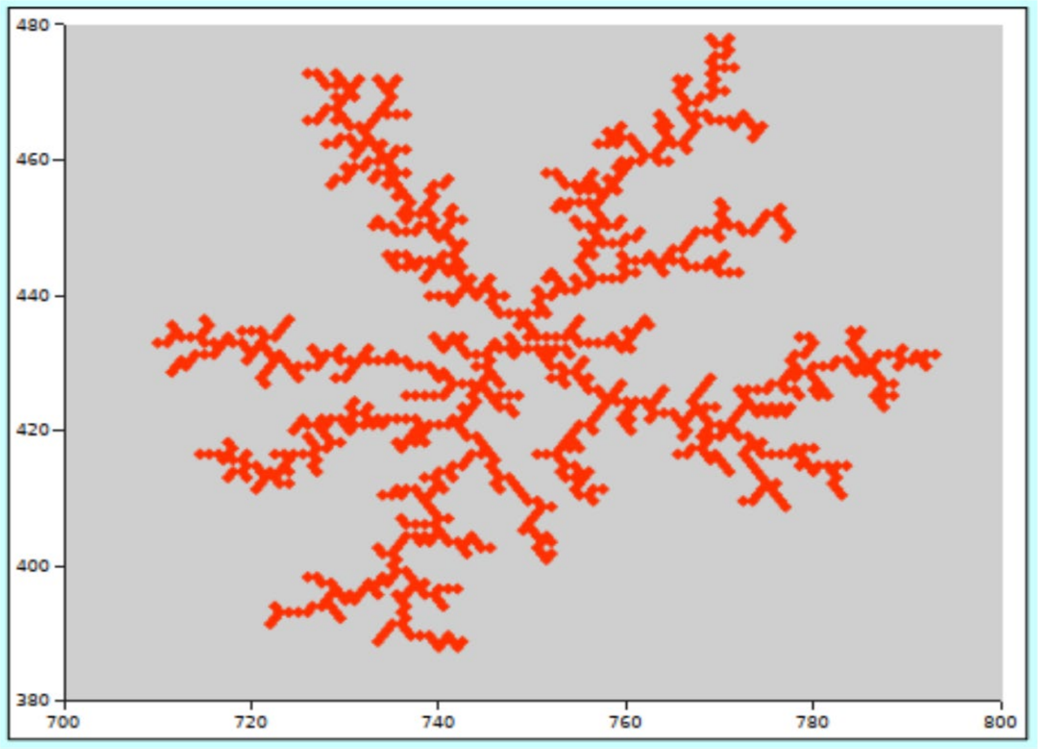
Snowflake simulation



```
#include <fstream>
#include <cmath>
#include <stdio.h>
#define pi acos(-1.0)
using namespace std;
int main (){
    ofstream out("snow.gnumeric");
    srand(time(NULL));
    int G=1000;
    int i,j;
    int a[G][G];
    for(i=0;i<G;i++){
        for(j=0;j<G;j++){
            a[i][j]=0;
        }
    }
    a[G/2][G/2]=1;
    int x,y;
int m;
int randommove;
int sample=100000,s;
for(s=0;s<sample;s++){

    m=rand()%4;
    switch(m){
        case 0:
            x=rand()%G;
            y=0;
            break;
        case 1:
            x=rand()%G;
            y=G-1;
            break;
        case 2:
            y=rand()%G;
            x=0;
            break;
        case 3:
            y=rand()%G;
            x=G-1;
            break;
    }

  while(1){
        randommove=rand()%6;
        switch(randommove){
            case 0:
                x++;
                x%=G;
                break;
            case 1:
                x--;
                x+=G;
                x%=G;
                break;
            case 2:
                y++;
                y%=G;
                break;
            case 3:
                y--;
               y+=G;
               y%=G;
                break;
            case 4:
                x++;
                x%=G;
                y--;
                y+=G;
                y%=G;
```

```
                    break;
                case 5:
                    x--;
                    x+=G;
                    x%=G;
                    y++;
                    y%=G;
                    break;
            }


            if(a[(x+1)%G][y]==1 or a[(x-1+G)%G][y]==1 or
a[x][(y+1)%G]==1 or
            a[x][(y-1+G)%G]==1  or a[(x-1+G)%G][(y+1)%G]==1 or
a[(x+1)%G][(y-1+G)%G]==1){
                a[x][y]=1;
                out<<x+y*0.5<<'\t'<<y*sqrt(3)/2<<'\n';
                break;
                 }
            }

    }

        }
```