**CPP1702 Coding Assignment 6**

HW6 計算機程式與應用

成大統計107級陳育婷 學號:H24034019

/\*

This code can be compiled and run ok.

usage (how to run):type input dataname,system will output input variables' values and initial Q0 grids' values.

If choose not to dose,enter 0, and system will out put grids' values of Q1,Q2,Q3,and Q0 in 1-5th iteration;

If choose to dose,enter number of dosing grids,row,column,culling rate, and affect radius(integers),

and system will out put grids' values of Q1,Q2,Q3,Q1+Q2+Q3,r(left over rate) and Q0 in 1-5th iteration.

input file: data1.txt

output file: none

compile (how to compile):

g++ -o hw6 hw6.cpp

pseudocode:

Part1:read file by function and output variables' values each grid's initial Q0 values by function

Part2 &3 :calculate growth(Q1)(\*1+s% if not given insecticide),spread(Q2)((grid k)\*q% for those grids within 1 Manhattan distance of k)if not given insecticide),

and pop-up(Q3)(randomly:70% 0,20% 40, 10% 60) by functions, and r(leftover rate=1-X(culling rate)\*2^d(distance R))in 1,2,3,4,5th iteration,

sum (Q1+Q2+Q3) times r to form Q0.

Output Q1,Q2,Q3, (Q1+Q2+Q3 and r if chose to dose)and Q0 for each iteration.

coded by Yu-Ting Chen, ID: H24034019, email: a8b3c5a@gmail.com

date: 2018.06.12\*/

//---begin--- PART 1: Read data1 and output read in variables, grid values in different colors, which are classified according to 4 value intervals ----------

#include <iostream>

#include <iomanip>

#include <fstream>

#include <string>

#include <cstdlib>

#include <cmath>

#include <ctime>

using namespace std ;

struct Zone{ //此為各網格之struct

int a; //儲存該網格之縱軸座標，即由上而下第a 列, a=0,…,M

int b; //儲存該網格之橫軸座標，即由左而右第b 行, b=0,…,N

int id; //儲存其編號，id=0,…,MN-1

float r=1.; //儲存該網格本期之殘存率，初始化為1，

//若該網格有投藥，則將有X 比例被撲殺，亦即殘留了1-X 比例。

//倘若另有一網格k’其所投放的藥效範圍會波及本網格，則會將網格

//k’對本網格的殘餘量（每多一單位則殺傷效果減半）拿來乘以

//本網格原先的殘餘量，以此類推。

bool isM; //儲存某網格於本期期初是否被投藥，是則為1，否則為0

};

struct Virus{

int F; //原生病毒出現之強度(或可理解為數量)

int p; //原生病毒該強度之發生機率(%)

};

int i,j,k,num,d;

void readFile(char \*,int &,int &,int &,int &,int &, int&,Zone \*&,Virus \*&,float \*&,float \*&,float \*&,float \*&);

void outputData(int ,int ,int ,int ,int , int ,Zone [],Virus [],float []);

void print\_colored\_2Darray(float [],int ,int,int,int);

void countQ1(int,Zone \*,float \*,float \*&);

void countQ2(int,Zone \*,float \*,float \*&);

void countQ3(Virus \*,float \*&);

void calculr(Zone \*&,int ,int);

void print\_mono\_2Darray(float \*,int ,int,int,int);

void print\_mono\_2Darraysum(float \*,float \*,float \*,int ,int ,int);

void print\_r(Zone \*,int ,int,int);

int main(){

int M, //區域縱軸座標範圍(1,2,…,M)

N, //區域橫軸座標範圍(1,2,…,N)

n\_Vir, //原生病毒出現之種類個數，譬如圖二(b)之n\_Vir=3 代表3 種

//病毒強度，分別為60,40,0 機率10%,20%,70%

T, //模擬之總期數

s, //給定的病毒每期成長比率(%)

q; //給定的病毒每期擴散比率(%)

Zone \*V; //動態宣告MN 長度之網格struct 陣列;V[k],k=0,1,…,MN-1

Virus \*Vir; //動態宣告n\_Vir 長度之原生病毒陣列;

float \*Q0,//Q0[k],k=0,…,MN-1 存網格k 於本期期末總病毒數量Q0=(Q1+Q2+Q3)\*r

\*Q1,//Q1[k]存k 之上期Q0 於本期成長所衍生的總病毒數量Q1=Q0\*(1+s)

\*Q2,//Q2[k]存k 之相鄰網格上期Q0 於本期擴散移入之病毒總量Q2+=鄰格Q0\*q

\*Q3;//Q3[k]儲存網格k 經過本期而新隨機出現的原生病毒數量，為某個強度F

int n\_M, //記錄當期投藥次數，若為0 則代表不投藥

row, //記錄當下投藥之網格縱軸座標

col, //記錄當下投藥之網格橫軸座標

X, //記錄當下投藥之撲殺率(%)

R; //記錄當下投藥之影響範圍，以曼哈頓距離表示，若為0 則代表僅影響那一格

char filename[50];//記錄輸入之測試檔名

int times=-1;

readFile(filename,M,N,T,s,q,n\_Vir,V,Vir,Q0,Q1,Q2,Q3);

outputData(M,N,T,s,q,n\_Vir,V,Vir,Q0);

print\_colored\_2Darray(Q0,M,N,times,0);

times=0;

while(times<5){

for (j=0;j<24;j++){

V[j].isM=0;

V[j].r=1;

}

cout<<"Enter Dosing Times in "<<times+1<<"th period:"<<"\n";

cin>>n\_M;

if(n\_M!=0){

for(j=0;j<n\_M;j++){

cout<<"Enter dosing row:"<<"\n";

cin>>row;

cout<<"Enter dosing column:"<<"\n";

cin>>col;

cout<<"Enter Culling Rate:"<<"\n";

cin>>X;

cout<<"Enter Affect Radius:"<<"\n";

cin>>R;

V[row\*N+col].isM=1;

}

}

countQ1(s,V,Q0,Q1);

countQ2(q,V,Q0,Q2);

countQ3(Vir,Q3);

calculr(V,X,R);

for(k=0;k<24;k++){

Q0[k]=(Q1[k]+Q2[k]+Q3[k])\*V[k].r;

}

print\_mono\_2Darray(Q1,M,N,times,1);

print\_mono\_2Darray(Q2,M,N,times,2);

print\_mono\_2Darray(Q3,M,N,times,3);

if(n\_M!=0){

print\_mono\_2Darraysum(Q1,Q2,Q3,M,N,times);

print\_r(V,M,N,times);

}

print\_colored\_2Darray(Q0,M,N,times,0);

times++;

}

delete []Vir;

delete []V;

delete []Q0;

delete []Q1;

delete []Q2;

delete []Q3;

}

void readFile(char \*filename,int &M,int &N,int &T,int &s,int &q, int

&n\_Vir,Zone \*&V,Virus \*&Vir,float \*&Q0,float \*&Q1,float \*&Q2,float \*&Q3){

cout<<"Enter dataname:";

fstream file;

cin>>filename;

ifstream inClientFile(filename, ios::in);

inClientFile>>M>>N>>T>>s>>q>>n\_Vir;

Vir=new Virus[n\_Vir];

inClientFile>>Vir[0].F>>Vir[0].p>>Vir[1].F>>Vir[1].p>>Vir[2].F>>Vir[2].p;

V=new Zone[M\*N];

Q0=new float[M\*N];

Q1=new float[M\*N];

Q2=new float[M\*N];

Q3=new float[M\*N];

for(i=0;i<M\*N;i++){

V[i].b=i%N;

k=i;

V[i].a=(k-(V[k].b))/N;

V[i].id=i;

V[i].r=1;

V[i].isM=0;

inClientFile>>Q0[i];

}

}

void outputData(int M,int N,int T,int s,int q, int n\_Vir,Zone V[],Virus Vir[],float Q0[]) {

cout<<"M="<<M<<",N="<<N<<", T="<<T<<",s="<<s<<"%, q="<<q<<"%, n\_Vir="<<n\_Vir<<";F={"<<Vir[0].F<<","<<Vir[1].F<<","<<Vir[2].F<<"},p={"<<Vir[0].p<<","<<Vir[1].p<<","<<Vir[2].p<<"};";

cout<<"\n";

}

void print\_colored\_2Darray(float \*Q0,int M,int N,int times,int step){

string col[4]={"\x1b[;32;1m","\x1b[;33;1m","\x1b[;31;1m","\x1b[;35;1m"};

string reset="\x1b[0m";

cout<<"Q"<<step<<"[k,"<<times+1<<"]"<<"\n";

for (i=0;i<M;i++){

for(j=0;j<N;j++){

if(Q0[i\*N+j]<33)

cout<<col[0] <<setw(6)<<fixed<<setprecision(1)<<Q0[i\*N+j] <<reset<<flush;

else if(Q0[i\*N+j]<66)

cout<<col[1] <<setw(6)<<fixed<<setprecision(1)<<Q0[i\*N+j] <<reset<<flush;

else if(Q0[i\*N+j]<100)

cout<<col[2] <<setw(6)<<fixed<<setprecision(1)<<Q0[i\*N+j] <<reset<<flush;

else

cout<<col[3] <<setw(6)<<fixed<<setprecision(1)<<Q0[i\*N+j] <<reset<<flush;

}

cout<<"\n";

}

}

//--end--- PART 1: Read data1 and output read in variables, grid values in different colors, which are classified according to 4 value intervals ----------

//--begin--- PART 2&3: simulate growth,spread ,pop up of virus ,and the effect of insecticide----------

void countQ1(int s,Zone \*V,float \*Q0,float \*&Q1){

for (i=0;i<24;i++){

if(V[i].isM==0){

Q1[i]=abs(Q0[i])\*(1.+(s/100.));

}else{

Q1[i]=Q0[i];

}

}

}

void countQ2(int q,Zone \*V,float \*Q0,float \*&Q2){

for(i=0;i<24;i++){

Q2[i]=0;

}

for(j=0;j<24;j++){

if(V[j].isM==0){

for(k=0;k<24;k++){

d=abs(V[k].a-V[j].a)+abs(V[k].b-V[j].b);

if(d==1)

Q2[k]=Q2[k]+Q0[j]\*(q/100.);

}

}

}

}

void countQ3(Virus \*Vir,float \*&Q3){

srand( static\_cast<unsigned int>(time(NULL)) );

for(i=0;i<24;i++){

num=1+rand()%100;

if(num<=10)

Q3[i]=60;

else if(num<=30)

Q3[i]=40;

else

Q3[i]=0;

}

}

void calculr(Zone \*&V,int X,int R){

for (i=0;i<24;i++){

if(V[i].isM==1){

for(j=0;j<24;j++){

d=abs(V[i].a-V[j].a)+abs(V[i].b-V[j].b);

if(d==0)

V[j].r=(100.-X)/100.;

if(d>0 && d<=R)

V[j].r=V[j].r\*pow(((100.-X/2.)/100.),d);

}

}

}

}

void print\_mono\_2Darray(float \*Qn,int M,int N,int times,int step){

cout<<"Q"<<step<<"[k,"<<times+1<<"]"<<"\n";

for (i=0;i<M;i++){

for(j=0;j<N;j++){

cout<<setw(6)<<fixed<<setprecision(1)<<Qn[i\*N+j];

}

cout<<"\n";

}

}

void print\_mono\_2Darraysum(float \*Qn1,float \*Qn2,float \*Qn3,int M,int N,int times){

cout<<"Q1[k,"<<times+1<<"]"<<"+Q2[k,"<<times+1<<"]"<<"+Q3[k,"<<times+1<<"]"<<"\n";

for (i=0;i<M;i++){

for(j=0;j<N;j++){

cout<<setw(6)<<fixed<<setprecision(1)<<Qn1[i\*N+j]+Qn2[i\*N+j]+Qn3[i\*N+j];

}

cout<<"\n";

}

}

void print\_r(Zone \*V,int M,int N,int times){

cout<<"r[k,"<<times+1<<"]"<<"\n";

for (i=0;i<M;i++){

for(j=0;j<N;j++){

cout<<setw(6)<<fixed<<setprecision(1)<<V[i\*N+j].r\*100.;

}

cout<<"\n";

}

}

//--end----- PART 2&3: simulate growth,spread ,pop up of virus ,and the effect of insecticide----------

**Output of Cygwin**



