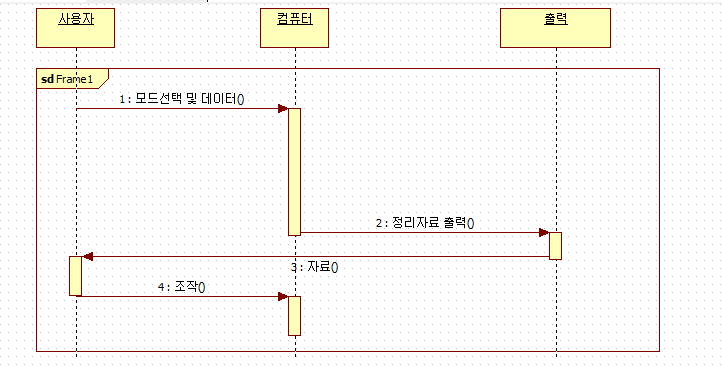
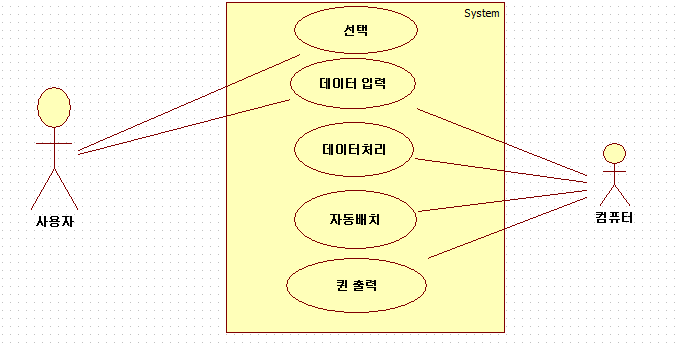
**N-퀸 문제 결과보고서**

20120302 김우진

**최종 실행 흐름도**

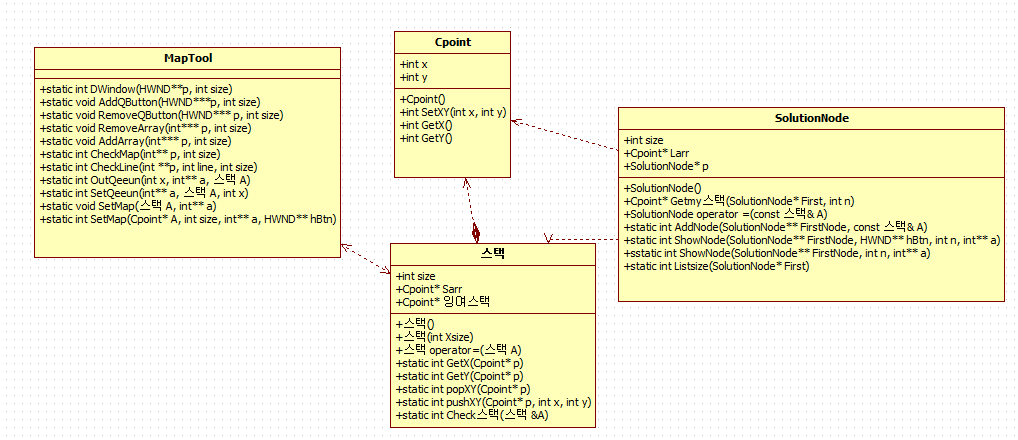
**<시스템 구성도와 결과 흐름도>**



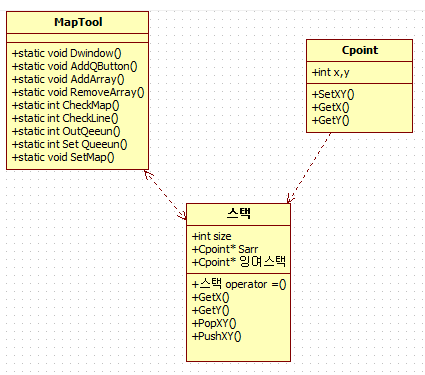


**<모듈 설계 및 클래스 입출력 >**

결과



설계



**주요 알고리즘에 대한 구현 코드 (알고리즘과 코드 비교)**

int MapTool::OutQeeun(int x,int\*\* a,스택 A){

while(스택::GetX(A.Sarr)!=스택::GetX(A.잉여스택)){

스택::PopXY(A.Sarr);

MapTool::SetMap(A,a);

}

스택::PopXY(A.Sarr);

스택::PushXY(A.Sarr,스택::GetX(A.잉여스택),스택::GetY(A.잉여스택));

스택::PopXY(A.잉여스택);

MapTool::SetMap(A,a);

return 스택::GetX(A.Sarr);

}

//퀸 추가함수 푸쉬를 호출함

int MapTool::SetQeeun(int\*\* p,스택 A,int x)

{

for(int i= A.size; i>=0; i--){

if(p[x][i]==0){

스택::PushXY(A.잉여스택,x,i);

}

}

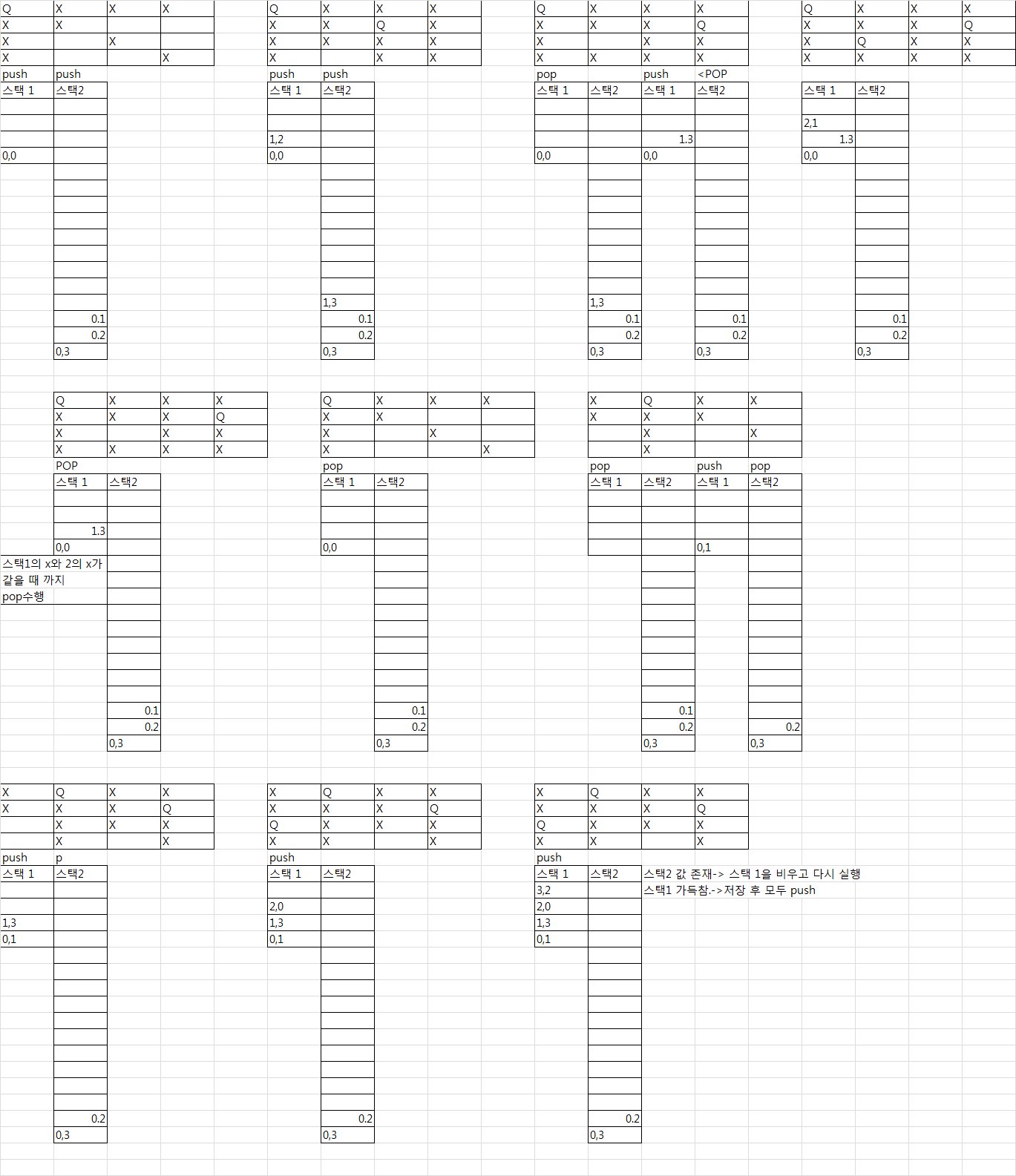
스택::PushXY(A.Sarr,스택::GetX(A.잉여스택), 스택::GetY(A.잉여스택));

스택::PopXY(A.잉여스택);

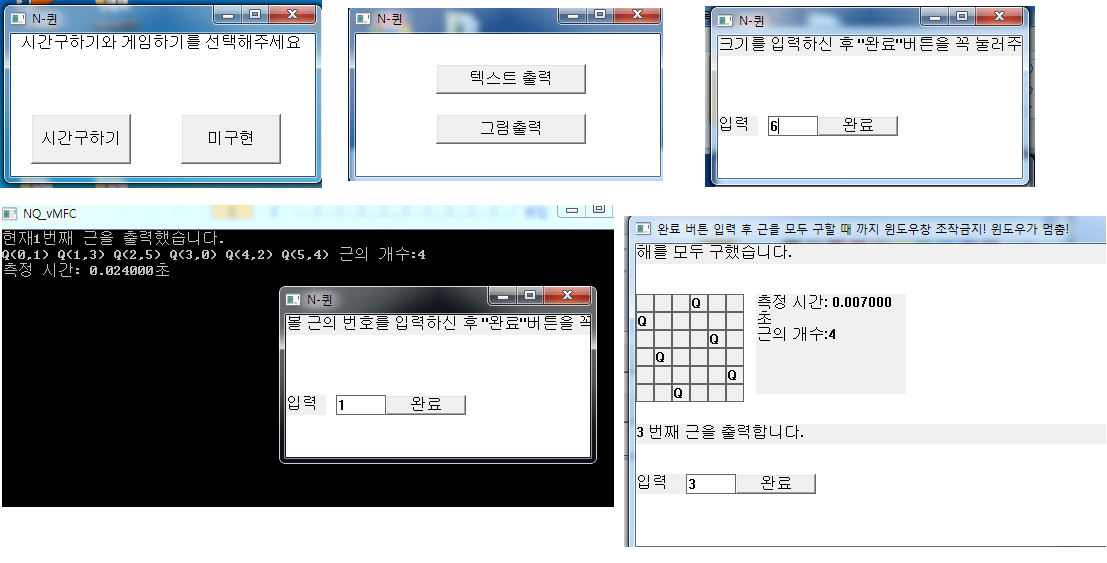
MapTool::SetMap(A,p);

return 0;

}



**개발 프로그램의 입출력 화면 및 실행 화면**



**오류 처리 종류 및 방법**

발견된 오류는 근을 몇 개 구하지 못하는 점과 근을 구하는 중 다른 조작을 하면 윈도우가 멈추는 것인데 처리 방법은 생각 하였으나 시간이 부족하여 구하지 못했습니다

\*다른 조작 시 멈춤: 스레드 사용

\*구하지 못하는 근: 백트레킹 외의 알고리즘 사용.

**느낀 점**

이번 소스코드에서 윈도우 프로그래밍과 스택을 사용하는 등 많은 것을 공부 하였고 배웠지만 가장 크게 느낀 것은 시간 복잡도와 공간 복잡도의 중요성 이였습니다. 처음에는 컴퓨터가 퀸을 옮길 때 마다 출력을 하여서 실시간으로 퀸을 볼 수 있게 하였으나 너무 빨라서 의미가 없고 오히려 시간만 잡아 먹어 너무 느렸습니다. 그래서 1번 근을 완성하면 1번근을 출력하고 나머지는 출력하지 중간 과정을 생략했더니 상상이상의 효과를 보았습니다. 예를 들자면, 초기 단계에서의 11ize의 시간은 1344초 가량이나 현재의 시간은 2.6 초로 거의 600배 이상의 속도가 증가했습니다. 앞으로는 프로그램의 작동도 중요하지만 효율성도 따져야 한다는 것을 크게 배웠고 느꼇습니다.

**전체 소스 코드**

- 충분한 주석을 포함

<스택.h>

#ifndef 스택\_H

#define 스택\_H

#endif // !스택\_H

#include<stdio.h>

#include<Windows.h>

#include<iostream>

using namespace std;

//좌표객체

class Cpoint{

public:

int x;

int y;

Cpoint(): x(0),y(0){};

int setXY(int a, int b);

int GetX(){return x;}

int GetY(){return y;}

};

class 스택{

public:

int size;

Cpoint\* Sarr;

Cpoint\* 잉여스택;

//생성자

스택(){size=0;Sarr=0; 잉여스택=NULL;};

스택(int xsize){

size = xsize; Sarr = new Cpoint [size+1]; 잉여스택 = new Cpoint [(size\*size)];

for(int i=0; i<size+1; i++)

Sarr[i].setXY(-1,-1);

for(int i=0; i<size\*size; i++)

잉여스택[i].setXY(-1,-1);

}

스택 operator =(스택 A){

size = A.size; Sarr =new Cpoint [size];

for(int i=0; i<size; i++)

Sarr[i] = A.Sarr[i];

return \*this;

}

//내부함수

static int GetX(Cpoint\* p){

int i=0;

for(i=0; ;i++)if(p[i].x==-1 &&p[i].y==-1)break;

if(i-1>=0) return p[i-1].x;

else return p[0].x;

}

static int GetY(Cpoint\* p){

int i=0;

for(i=0; ;i++) if(p[i].x==-1 &&p[i].y==-1)break;

if(i-1>=0)return p[i-1].y;

return p[0].y;

}

static void PopXY(Cpoint\* p){

int i=0;

for(i=0; ;i++)if(p[i].x==-1 &p[i].y==-1) break;

if(i-1>=0) p[i-1].setXY(-1,-1);

else p[0].setXY(-1,-1);

}

static void PushXY(Cpoint\* p ,int x,int y){

int i=0;

for(i=0; ;i++) if(p[i].x==-1 &&p[i].y==-1) break;

if(i-1>=0) p[i].setXY(x,y);

else p[0].setXY(x,y);

}

static int Check스택(스택 &A){

int size=A.size; int i=0;

for(i=0;i<size\*size ;i++) if(A.잉여스택[i].x !=-1 || A.잉여스택[i].y!=-1) return 0; //하나라도 덜 빠졌다면.

if(i==size\*size) return 1; //스택이 비었다면

//위의 두 경우가아니라면 오류 출력

else return -1;

}

};

//맵 버튼 관리

class MapTool{

public:

//버튼과 맵생성 함수

static void DWindow(HWND\*\* p,int size);

static void AddQButton(HWND\*\*\* p,int size);

static void RemoveQButton(HWND\*\*\* p,int size);

static void RemoveArray(int\*\*\* p,int size);

static void AddArray(int\*\*\* p,int size);

//맵관리 함수

static int CheckMap(int\*\* p,int size);

static int CheckLine(int \*\*p,int line,int size); //라인에 퀸배치가 가능:1, 라인에 퀸 존재 0, 배치불가 -1

static int OutQeeun(int x,int\*\* a,스택 A);

static int SetQeeun(int\*\* a, 스택 A,int x);

static void SetMap(스택 A,int\*\* a);

static int SetMap(Cpoint\* A,int size,int\*\* a,HWND\*\* hBtn);

};

// 해 관리

class SolutionNode{

public:

int size;

Cpoint\* Larr;

SolutionNode\* p;

SolutionNode(){

size=0;

Larr =NULL;

p=NULL;

}

Cpoint\* Getmy스택(SolutionNode\* First,int n){

SolutionNode\* p= First;

for(int i=0; i<n;i++){p=p->p; }

return p->Larr;

}

SolutionNode operator =(const 스택& A){

this->size=A.size;

this->Larr=new Cpoint[size];

for(int i=0;i<size;i++)

Larr[i]=A.Sarr[i];

return \*this;

}

//리스트 추가함수

static int AddNode(SolutionNode\*\* FirstNode,const 스택& A){

SolutionNode\* New=new SolutionNode();

\*New=A;

if(!(\*FirstNode) ){ \*FirstNode = New;return 0;}

SolutionNode\* p = \*FirstNode;

while(p->p) p=p->p;

p->p=New;

return 0;

}

//리스트 출력함수

static int ShowNode(SolutionNode\*\* FirstNode,HWND\*\* hBtn,int n,int\*\* a){

if(!(\*FirstNode)) return 0;

SolutionNode\* p=\*FirstNode;

for(int i=0; i<n; i++){

p=p->p;

}

int size=p->size;

for(int m=0; m<size; m++){

for(int n=0;n<size;n++){

a[m][n]=0;

}

}

for(int i=0;i<size;i++){

int x=p->Larr[i].x; int y=p->Larr[i].y;

if(x==-1&&y==-1)break;

for(int m=0; m<size; m++){

for(int n=0;n<size;n++){

if(m-n==x-y || m+n==x+y) a[m][n]=-1;

if(m==x || n==y)a[m][n]=-1;

}

}

for(int m=0; m<size; m++){

int x=p->Larr[i].x; int y=p->Larr[i].y;

if(x==-1&&y==-1)break;

a[x][y]=1;

}

}

//텍스트박스 업데이트

for(int i= 0; i<size; i++){

for(int j=0;j<size;j++){

if(a[i][j]==1)

SetWindowText(hBtn[i][j],"Q");

else

SetWindowText(hBtn[i][j]," ");

}

}

return 0;

}

static int ShowNode(SolutionNode\*\* FirstNode,int n,int\*\* a){

system("cls");

if(!(\*FirstNode)) return 0;

SolutionNode\* p=\*FirstNode;

for(int i=0; i<n; i++){

p=p->p;

}

cout<<"현재"<<n+1<<"번째 근을 출력했습니다."<<endl;

int size =(\*FirstNode)->size;

char\*\* C= new char\*[(\*FirstNode)->size];

for(int i=0; i<size; i++){

if(p->Larr[i].GetX()==-1 && p->Larr[i].GetY()==-1) break;

printf("Q(%d,%d) ",p->Larr[i].GetX(),p->Larr[i].GetY());

if(i%8==0&& i!=0) cout<<endl;

}

p= \*FirstNode;

cout<<"근의 개수:"<<SolutionNode::Listsize(p);

cout<<endl;

return 0;

}

static int Listsize(SolutionNode\* First){

SolutionNode\* p =First;

int count=0;

while(1){

if(p==NULL) break;

count++;

p=p->p;

}

return count;

}

};

<스택.cpp>

#include"스택.h"

#include<Windows.h>

#include<stdio.h>

#include <iostream>

using namespace std;

int Cpoint::setXY(int a, int b){

x=a; y=b;

return 0;

}

int MapTool::CheckMap(int\*\* p,int size){

for(int i=0;i<size;i++){

//라인에 퀸이 없다면

if( MapTool::CheckLine(p,i,size)!=0) return 1;

}//모든 라인에 퀸 존재

return 0;

}

//퀸을 빼며 팝을 호출하는 함수

int MapTool::OutQeeun(int x,int\*\* a,스택 A){

while(스택::GetX(A.Sarr)!=스택::GetX(A.잉여스택)){

스택::PopXY(A.Sarr);

MapTool::SetMap(A,a);

}

스택::PopXY(A.Sarr);

스택::PushXY(A.Sarr,스택::GetX(A.잉여스택),스택::GetY(A.잉여스택));

스택::PopXY(A.잉여스택);

MapTool::SetMap(A,a);

return 스택::GetX(A.Sarr);

}

//퀸 추가함수 푸쉬를 호출함

int MapTool::SetQeeun(int\*\* p,스택 A,int x)

{

for(int i= A.size; i>=0; i--){

if(p[x][i]==0){

스택::PushXY(A.잉여스택,x,i);

}

}

스택::PushXY(A.Sarr,스택::GetX(A.잉여스택), 스택::GetY(A.잉여스택));

스택::PopXY(A.잉여스택);

MapTool::SetMap(A,p);

return 0;

}

//

void MapTool::DWindow(HWND\*\* p,int size){

for(int i= 0; i<size; i++){

for(int j=0;j<size;j++){

DestroyWindow(p[i][j]);

}

}

}

void MapTool::AddQButton(HWND\*\*\* p,int size){

\*p = new HWND\* [size];

for(int i=0; i<size; i++){

(\*p)[i] = new HWND [size];

}

}

void MapTool::AddArray(int\*\*\* p,int size){

\*p = new int\* [size];

for(int i=0; i<size; i++){

(\*p)[i] = new int [size];

}

for(int i=0; i<size; i++)

for(int j=0; j<size;j++)

(\*p)[i][j]=0;

}

void MapTool::RemoveQButton(HWND\*\*\* p,int size){

for(int i=0; i<size; i++)

delete [] (\*p[i]);

delete [] p;

}

void MapTool::RemoveArray(int\*\*\* p,int size){

for(int i=0; i<size; i++)

delete [] (\*p[i]);

delete [] p;

}

//주어진 라인의 상태 체크

int MapTool:: CheckLine(int \*\*p,int line,int size){

//한 곳이라도 놓을 곳이 있다면 1 반환

for(int i=0; i<size; i++){

if(p[line][i]==0) {

return 1;}

}

//배치할 곳이 없다.

int i=0;

for(i=0; i<size; i++)

if(p[line][i]==1){return 0;}//퀸O 배치불가

return -1;//라인의 퀸X 배치불가

}

int MapTool::SetMap(Cpoint\* A,int size,int\*\* a,HWND\*\* hBtn){

for(int m=0; m<size; m++){

for(int n=0;n<size;n++){

a[m][n]=0;

}

}

for(int i=0;i<size;i++){

int x=A[i].x; int y=A[i].y; if(x==-1&&y==-1)break;

for(int m=0; m<size; m++){

for(int n=0;n<size;n++){

if(m-n==x-y || m+n==x+y) a[m][n]=-1; if(m==x || n==y)a[m][n]=-1;

}

}

for(int m=0; m<size; m++){

int x=A[i].x; int y=A[i].y; if(x==-1&&y==-1)break;

a[x][y]=1;

}

}

return 0;

}

// 퀸을 배치, 공격범위를 표시

void MapTool::SetMap(스택 A,int\*\* a){

int size=A.size;

for(int m=0; m<size; m++){

for(int n=0;n<size;n++){

a[m][n]=0;

}

}

for(int i=0;i<size;i++){

int x=A.Sarr[i].x; int y=A.Sarr[i].y; if(x==-1&&y==-1)break;

for(int m=0; m<size; m++){

for(int n=0;n<size;n++){

if(m-n==x-y || m+n==x+y) a[m][n]=-1;

else if(m==x || n==y)a[m][n]=-1;

}

}

}

for(int m=0; m<size; m++){

int x=A.Sarr[m].x; int y=A.Sarr[m].y;

if(x==-1&&y==-1)break;

a[x][y]=1;

}

}

<winmain>

#include <iostream>

#include<Windows.h>

#include "스택.h"

#include<time.h>

#include<stdio.h>

using namespace std;

//#pragma comment(linker, "/entry:WinMainCRTStartup /subsystem:console")

//함수선언(<=>프로토타입)

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

LRESULT CALLBACK WndProc2(HWND, UINT, WPARAM, LPARAM);

LRESULT CALLBACK WndProc3(HWND, UINT, WPARAM, LPARAM);

int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpszArg, int nCmdShow)

{

HWND hWnd;

MSG msg;

WNDCLASS WndClass;

char szAppName[] ="N-퀸";

WndClass.style = NULL;

WndClass.lpfnWndProc = WndProc;

WndClass.cbClsExtra = 0;

WndClass.cbWndExtra = 0;

WndClass.hInstance = hInstance;

WndClass.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

WndClass.hCursor = LoadCursor(NULL, IDC\_ARROW);

WndClass.hbrBackground = (HBRUSH) GetStockObject(WHITE\_BRUSH);

WndClass.lpszMenuName = NULL;

WndClass.lpszClassName = szAppName;

if(!RegisterClass(&WndClass)) return NULL;

WndClass.lpszClassName = "자식1";

WndClass.lpfnWndProc = WndProc2;

if(!RegisterClass(&WndClass)) return NULL;

hWnd = CreateWindow(

szAppName,

szAppName,

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

320,

180,

NULL,

NULL,

hInstance,

NULL

);

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

while(GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return msg.wParam;

}

LRESULT CALLBACK WndProc(HWND hWnd, UINT mesg, WPARAM wParam, LPARAM lParam)

{

static HWND son;

static HDC hdc = GetDC(hWnd);

static int\*\* a=NULL; //맵변수. x,y에 q의 유무에 따라 0,1,-1 등을 저장

static int size =8;//기본 사이즈

static int id = 0; //버튼 구분변수

//ingame, id로부터버튼의 좌표를 받아오는 변수

int x= 0;

int y= 0;

static Cpoint\* QPoint=NULL;

char s[100];

static HINSTANCE \_hInstance = (HINSTANCE)GetWindowLong(hWnd,GWL\_HINSTANCE);

static HWND\*\* hBtn = NULL; //sizeXsize의 버튼을 만들 포인터. 동적할당

static HWND Bt,up,down,start,s1,s2,ok;//버튼, up/down 설정버튼, 시작버튼 , 선택1~2버튼,확인버튼

static int line=0;

static SolutionNode\* First=NULL;

/\* 스위치(이벤트처리) 시작\*/

switch(mesg)

{

case WM\_CREATE:

Bt =CreateWindow("EDIT","시간구하기와 게임하기를 선택해주세요",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,10,0,300,30,hWnd,NULL,\_hInstance,NULL);

s1=CreateWindow("BUTTON","시간구하기",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,20,80,100,50,hWnd,(HMENU)1,\_hInstance,NULL);

s2=CreateWindow("BUTTON","미구현",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,170,80,100,50,hWnd,(HMENU)2,\_hInstance,NULL);

static HWND TextBoxA,TextBoxB;

break;

//버튼 누름 이벤트처리

case WM\_COMMAND:

id = LOWORD(wParam);

//시간 측정을 택했다면

if(id==1){

DestroyWindow(Bt);DestroyWindow(s1);DestroyWindow(s2);

s1 =CreateWindow("BUTTON","텍스트 출력",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,80,30,150,30,hWnd,(HMENU)3,\_hInstance,NULL);

s2 =CreateWindow("BUTTON","그림출력",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,80,80,150,30,hWnd,(HMENU)4,\_hInstance,NULL);

}

//그림 선택이라면

if(id==4){

son=CreateWindow("자식1","완료 버튼 입력 후 근을 모두 구할 때 까지 윈도우창 조작금지! 윈도우가 멈춤!", WS\_OVERLAPPEDWINDOW, CW\_USEDEFAULT, CW\_USEDEFAULT,500,150, hWnd, NULL, \_hInstance, NULL );

ShowWindow(son,SW\_SHOW);

DestroyWindow(s1);DestroyWindow(s2);

}

//텍스트를 택했다면

if(id==3){

DestroyWindow(s1);DestroyWindow(s2);

TextBoxA =CreateWindow("EDIT","크기를 입력하신 후 \"완료\"버튼을 꼭 눌러주세요",WS\_VISIBLE|WS\_CHILD|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER

|ES\_READONLY,0,0,500,20,hWnd,NULL,\_hInstance,NULL);

TextBoxB =CreateWindow("EDIT","입력",WS\_VISIBLE|WS\_CHILD|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER|ES\_READONLY,0,80,40,20,hWnd,NULL,\_hInstance,NULL);

Bt =CreateWindow("EDIT","",WS\_VISIBLE|WS\_CHILD|ES\_AUTOHSCROLL|ES\_AUTOVSCROLL|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER

|WS\_BORDER,50,80,50,20,hWnd,NULL,\_hInstance,NULL);

ok =CreateWindow("BUTTON","완료",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,100,80,80,20,hWnd,(HMENU)10,\_hInstance,NULL);

}

//텍스트 선택 -> 사이즈 입력 완료

if(id==10){

GetWindowText(Bt, s, 100);//입력한 사이즈를 받음

size=atoi(s);//정수화

if(size<4) size=4;//최솟값 미만의 값일 경우 최솟값으로

//필요없는 버튼 파괴

DestroyWindow(Bt);DestroyWindow(ok);DestroyWindow(TextBoxB);

//콘솔 실행

AllocConsole();

AttachConsole(GetCurrentProcessId());

freopen( "CONOUT$", "w+", stdout ) ;

std::ios::sync\_with\_stdio();

DestroyWindow(s1);DestroyWindow(s2);

MapTool::AddArray(&a,size);

static 스택 A(size);static clock\_t start=clock(); //사용할 스택 등록, 시간 측정시작

static int index =0;

//자동배치 시작

//라인 별로 퀸을 배치

while(1){

static int line =0;

if(line>size-1) line=size-1;

if(MapTool::CheckLine(a,line,size)==1){

MapTool::SetQeeun(a,A,line); line++;

}

else{line--;

line=1+MapTool::OutQeeun(line,a,A);

}

if(!(MapTool::CheckMap(a,size))){

SolutionNode::AddNode(&First,A);

if(!index){

SolutionNode::ShowNode(&First,0,a);

if(size>=13) break;

index++;

}

}

//잉여스택이 비었다면 종료

if(스택::Check스택(A)){break;}

}

//뒷처리

clock\_t end=clock();

double time = (double)(end-start)/CLOCKS\_PER\_SEC;

SolutionNode::ShowNode(&First,0,a);

if(size<13)

printf("측정 시간: %lf초",time);

if(size>=13){

time =(100\*size-934)/3;//회귀분석법을 통해 얻은식.

time\*=1.1;

printf("예측 시간은 %lf초 입니다",time);

}

TextBoxA =CreateWindow("EDIT","볼 근의 번호를 입력하신 후 \"완료\"버튼을 꼭 눌러주세요",WS\_VISIBLE|WS\_CHILD|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER

|ES\_READONLY,0,0,500,20,hWnd,NULL,\_hInstance,NULL);

TextBoxB =CreateWindow("EDIT","입력",WS\_VISIBLE|WS\_CHILD|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER|ES\_READONLY,0,80,40,20,hWnd,NULL,\_hInstance,NULL);

Bt =CreateWindow("EDIT","",WS\_VISIBLE|WS\_CHILD|ES\_AUTOHSCROLL|ES\_AUTOVSCROLL|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER

|WS\_BORDER,50,80,50,20,hWnd,NULL,\_hInstance,NULL);

ok =CreateWindow("BUTTON","완료",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,100,80,80,20,hWnd,(HMENU)30,\_hInstance,NULL);

};

// 텍스트의 결과에서 선택한 근을 출력

if(id==30){

GetWindowText(Bt, s, 100);

int num=atoi(s)-1;

if(0>num) num=0;

else if(num>=SolutionNode::Listsize(First)) num=SolutionNode::Listsize(First) -1;

SolutionNode::ShowNode(&First,num,a);

}

break;

case WM\_DESTROY :

PostQuitMessage(0);

return FALSE;

}

return DefWindowProc(hWnd, mesg, wParam, lParam);

}

LRESULT CALLBACK WndProc2(HWND hWnd, UINT mesg, WPARAM wParam, LPARAM lParam)

{

static HDC hdc = GetDC(hWnd);

static int\*\* a=NULL; //맵변수. x,y에 q의 유무에 따라 0,1,-1 등을 저장

static int size =8;//기본 사이즈

static int id = 0; //버튼 구분변수

//ingame, id로부터버튼의 좌표를 받아오는 변수

int x= 0;

int y= 0;

static Cpoint\* QPoint=NULL;

char s[100];

static HINSTANCE \_hInstance = (HINSTANCE)GetWindowLong(hWnd,GWL\_HINSTANCE);

static HWND\*\* hBtn = NULL; //sizeXsize의 버튼을 만들 포인터. 동적할당

static HWND Bt,up,down,start,s1,s2,ok;//버튼, up/down 설정버튼, 시작버튼 , 선택1~2버튼,확인버튼

static int line=0;

static SolutionNode\* First=NULL;

static HWND TextBoxA,TextBoxB;

static int index=0;

/\* 스위치(이벤트처리) 시작\*/

switch(mesg)

{

case WM\_CREATE:

TextBoxA =CreateWindow("EDIT","크기를 입력하신 후 \"완료\"버튼을 꼭 눌러주세요",WS\_VISIBLE|WS\_CHILD|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER

|ES\_READONLY,0,0,500,20,hWnd,NULL,\_hInstance,NULL);

TextBoxB =CreateWindow("EDIT","입력",WS\_VISIBLE|WS\_CHILD|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER|ES\_READONLY,0,80,40,20,hWnd,NULL,\_hInstance,NULL);

Bt =CreateWindow("EDIT","",WS\_VISIBLE|WS\_CHILD|ES\_AUTOHSCROLL|ES\_AUTOVSCROLL|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER

|WS\_BORDER,50,80,50,20,hWnd,NULL,\_hInstance,NULL);

ok =CreateWindow("BUTTON","완료",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,100,80,80,20,hWnd,(HMENU)10,\_hInstance,NULL);

break;

case WM\_COMMAND:

id = LOWORD(wParam);

if(id==10){

SetWindowText(TextBoxA,"해를 모두 구했습니다.");

//사이즈조절

GetWindowText(Bt, s, 100);

size=atoi(s);

if(size<4) size=4;

MoveWindow(hWnd,0,0,size\*18+500,size\*20+220,true);

//필요없는 버튼 파괴

DestroyWindow(Bt);DestroyWindow(ok);DestroyWindow(TextBoxB);

MapTool::AddQButton(&hBtn,size);

MapTool::AddArray(&a,size);

//버튼그리기 시작

for(int i= 0; i<size; i++){

for(int j=0;j<size;j++){

if(a[i][j]==1)

hBtn[i][j]= CreateWindow("Static", "Q",WS\_VISIBLE|WS\_CHILD|WS\_BORDER,j\*18, 18\*i+50, 18, 18, hWnd,NULL, \_hInstance, NULL);

else

hBtn[i][j]= CreateWindow("Static", " ",WS\_VISIBLE|WS\_CHILD|WS\_BORDER,j\*18, 18\*i+50, 18, 18, hWnd,NULL, \_hInstance, NULL);

}

}//버튼그리기 끝.

static 스택 A(size);static clock\_t start=clock(); //사용할 스택 등록, 시간 측정시작

//자동배치 시작

while(1){

static int line =0;

if(line>size-1) line=size-1;

if(MapTool::CheckLine(a,line,size)==1){

MapTool::SetQeeun(a,A,line); line++;

}

else{line--;

line=1+MapTool::OutQeeun(line,a,A);

}

if(!(MapTool::CheckMap(a,size))){

SolutionNode::AddNode(&First,A);

if(!index){

SolutionNode::ShowNode(&First,hBtn,0,a);

if(size>=13) break;

}

index++;

}

if(스택::Check스택(A))break;

}//while 끝

//뒷처리

SolutionNode::ShowNode(&First,hBtn,0,a);

clock\_t end=clock();

double time = (double)(end-start)/CLOCKS\_PER\_SEC;

if(size<13)

sprintf(s,"측정 시간: %lf초\n근의 개수:%d",time,SolutionNode::Listsize(First));

if(size>=13){

time=(100\*size-934)/3;

time\*=1.1;

sprintf(s,"예측 시간: %lf초\n(예상오차10퍼센트)",time);

}

//구한 근을 보기

Bt =CreateWindow("STATIC",s,WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,20\*size,50,150,100,hWnd,0,\_hInstance,NULL);

TextBoxA =CreateWindow("EDIT","근의 번호를 입력 후 반드 시 \"완료\"버튼을 꼭 눌러주세요!",WS\_VISIBLE|WS\_CHILD|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER

|ES\_READONLY,0,25\*size+30,500,20,hWnd,NULL,\_hInstance,NULL);

TextBoxB =CreateWindow("EDIT","입력",WS\_VISIBLE|WS\_CHILD|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER|ES\_READONLY,0,25\*size+80,50,20,hWnd,NULL,\_hInstance,NULL);

Bt =CreateWindow("EDIT","",WS\_VISIBLE|WS\_CHILD|ES\_AUTOHSCROLL|ES\_AUTOVSCROLL|ES\_MULTILINE|ES\_NOHIDESEL|ES\_NUMBER

|WS\_BORDER,50,25\*size+80,50,20,hWnd,NULL,\_hInstance,NULL);

ok =CreateWindow("BUTTON","완료",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,100,25\*size+80,80,20,hWnd,(HMENU)20,\_hInstance,NULL);

}

if(id==20){

//사이즈조절

GetWindowText(Bt, s, 100);

int num=atoi(s)-1;

if(0>num) num=0;

else if(num>=SolutionNode::Listsize(First)) num=SolutionNode::Listsize(First) -1;

sprintf(s,"%d 번째 근을 출력합니다.",num+1);

SetWindowText(TextBoxA,s);

SolutionNode::ShowNode(&First,hBtn,num,a);

//필요없는 버튼 파괴

DestroyWindow(ok);

ok =CreateWindow("BUTTON","완료",WS\_VISIBLE|WS\_CHILD|BS\_PUSHBUTTON,100,25\*size+80,80,20,hWnd,(HMENU)20,\_hInstance,NULL);

}

break;

case WM\_DESTROY :

PostQuitMessage(0);

return FALSE;

}

return DefWindowProc(hWnd, mesg, wParam, lParam);

}