#include <iostream>

#include <random>

**enum** Color{*RED*, *YELLOW*, *BLUE*, *BLACK*};

**struct** Posicion{

**int** x;

**int** y;

Posicion(**int** x, **int** y) : x(x), y(y){}

Posicion(){}

};

**struct** Punto{

Posicion pos;

Color col;

Punto \*next;

Punto \*prev;

Punto(Posicion pos, Color col): pos(pos), col(col), next(**nullptr**), prev(**nullptr**){}

Punto() {}

};

**class** Pila{

Punto \*top;

**int** size;

**public**:

Pila() : top(**nullptr**), size(0){} // Initialize

**void** Destroy(){

Punto \*actual = top;

**do**{

top=actual->prev;

**delete** actual;

actual=top;

size--;

}**while**(actual!= **nullptr**);

}

**void** Push(Posicion \_pos, Color \_col){

Punto\* nuevo = **new** Punto(\_pos, \_col);

**if** (IsEmpty()){

top = nuevo;

}**else**{

top->next=nuevo;

nuevo->prev=top;

top=nuevo;

}

size++;

}

Punto Pop(){

Punto \*borrar = top;

Punto retornar(borrar->pos,borrar->col);

top=borrar->prev;

size--;

**delete** borrar;

**return** retornar;

}

Punto Peek(){

Punto retorn(top->pos,top->col);

**return** retorn;

}

**int** Size(){**return** size;}

**bool** IsEmpty(){**return** top == **nullptr**;}

Punto\* GetTop(){**return** top;}

**void** Print(){

Punto \*buscar = top;

**while**(buscar!= **nullptr**){

std::cout<<"[Punto: "<<"("<<buscar->pos.x<<", "<<buscar->pos.y<<") - "<<"Color: "<<buscar->col<<"]"<<std::endl;

buscar=buscar->prev;

}

std::cout<<std::endl;

}

};

**void** Agrupar(Pila \*original, Pila \*Red, Pila \*Yellow, Pila\* Blue, Pila \*Black){

Punto \*actual = original->GetTop();

**while**(actual!= **nullptr**){

**switch** (actual->col){

**case** 0:

Red->Push(actual->pos, actual->col);

**break**;

**case** 1:

Yellow->Push(actual->pos, actual->col);

**break**;

**case** 2:

Blue->Push(actual->pos, actual->col);

**break**;

**case** 3:

Black->Push(actual->pos, actual->col);

**break**;

}

actual=actual->prev;

}

}

**int** main() {

std::random\_device rand;

Color x;

Posicion p1;

**int** cant = 20;

//Color c1 = Color::RED;

//Punto \*pos = new Punto(p1,c1);

Pila \*pila = **new** Pila();

Pila \*red = **new** Pila();

Pila \*yellow = **new** Pila();

Pila \*blue = **new** Pila();

Pila \*black = **new** Pila();

**for** (**int** i = 0; i < cant ; ++i) {

x = Color (rand()%4);

p1 = Posicion((rand()%601),(rand()%601));

pila->Push(p1,x);

}

pila->Print();

Agrupar(pila, red, yellow, blue, black);

std::cout<<"Solo Rojos: "<<std::endl;

red->Print();

std::cout<<"Solo Amarillos: "<<std::endl;

yellow->Print();

std::cout<<"Solo Azules: "<<std::endl;

blue->Print();

std::cout<<"Solo Negros: "<<std::endl;

black->Print();

pila->Destroy();

red->Destroy();

yellow->Destroy();

blue->Destroy();

black->Destroy();

**delete** pila;

**delete** red;

**delete** yellow;

**delete** blue;

**delete** black;

**return** 0;

}