**dUSet.h**

#ifndef DUSET\_H

#define DUSET\_H

#include "dynamicBag.h"

#include <string>

class dUSet: public dynamicBag{

public:

// default constructor

// pre: none

// post: creates an empty dynamicBag

dUSet();

// constructor that creates a dUSet out of an array of integers

// pre: size must match the size of the array

// post: creates a new dUSet which is a copy of the array passed in

dUSet(int a[], int sz);

// pre: none

// post: adds a copy of target to this dynamicBag if, and only if,

//the int passed in doesn't exist in the set

void insert(int target);

};

#endif

**dUSet.cpp**

#include "dUSet.h"

dUSet::dUSet(): dynamicBag(){

}

dUSet::dUSet(int a[], int sz): dynamicBag(a, sz){

int numToErase;

size\_t numItemsErased = 0;

for(int i = 0; i < sz-1; i++){

for(int j = i+1; j < sz; j++){

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

sz--;

numToErase = a[i];

for(int k = i; k < sz; k++){

a[k] = a[k+1];

}

numItemsErased = erase\_one(numToErase);

j--;

i--;

}

}

}

}

void dUSet::insert(int target){

erase(target);

dynamicBag::insert(target);

}

**main.cpp | Only for Problem 2(dUSet)**

#include "dUSet.h"

#include "dSet.h"

#include <iostream>

using namespace std;

int main(){

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"Testing Problem 2, Constructor with array parameter"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<""<<endl;

int arr[10] = {1,3,2,6,3,2,1,6,2,6};

dUSet firstDUSet = dUSet(arr, 10);

cout<<"Array we start with: ";

for(int i = 0; i < 10; i++){

cout<<arr[i] << ", ";

}

cout<<""<<endl;

cout<<"This is the set that automatically erases duplicates: ";

cout<<firstDUSet<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"Testing Problem 2, overloaded insert function"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<""<<endl;

cout << "Set we start with: " << firstDUSet << endl;

cout << "Inserting a number that exists in the set (3): ";

firstDUSet.insert(3);

cout << firstDUSet <<endl;

cout << "Inserting a number that does NOT exist in the set (4): ";

firstDUSet.insert(4);

cout << firstDUSet <<endl;

cout<<""<<endl;

cout<<""<<endl;

return(0);

}

**Result from running main**

FHosts-MacBook-Pro:dyanmicBag fhost$ g++ \*.cpp -o main

FHosts-MacBook-Pro:dyanmicBag fhost$ ./main

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Testing Problem 2, Constructor with array parameter

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Array we start with: 3, 1, 2, 6, 6, 6, 6, 6, 6, 6,

This is the set that automatically erases duplicates: 3 1 2 6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Testing Problem 2, overloaded insert function

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Set we start with: 3 1 2 6

Inserting a number that exists in the set (3): 1 2 6 3

Inserting a number that does NOT exist in the set (4): 1 2 6 3 4