Worksheet 0: Building a Simple ADT Using an Array
In Preparation: Read about basic ADTs.

In this worksheet we will construct a simple BAG and STACK abstraction on top of an array. Assume we have the following interface file "arrayBagStack.h"

```
----------
# ifndef ArrayBagStack
# define ArrayBagStack
# define MAX_SIZE 100


# define TYPE int
# define EQ(a, b) (a == b)

struct arrayBagStack {
     TYPE data [MAX_SIZE];
     int count;
};

/* Interface for Bag */
void initBag (struct arrayBagStack * b);
void addBag (struct arrayBagStack * b, TYPE v);
int containsBag (struct arrayBagStack * b, TYPE v);
void removeBag (struct arrayBagStack * b, TYPE v);
int sizeBag (struct arrayBagStack * b);

/* Interface for Stack */
void pushStack (struct arrayBagStack * b, TYPE v);
TYPE topStack (struct arrayBagStack * b);
void popStack (struct arrayBagStack * b);
int isEmptyStack (struct arrayBagStack * b);

# endif
----------
```

Your job, for this worksheet, is to provide implementations for the following operations.

```
void initBag (struct arrayBagStack * b){

     b->count = 0;

}

void addBag (struct arrayBagStack * b, TYPE v) {
```

```
     assert(b->count != MAX_SIZE);
     b->data[b->count] = v;
     b->count += 1;

}

int containsBag (struct arrayBagStack * b, TYPE v) {

     for(int i = 0; i < b->count; i++){
          if(b->data[i] == v){
               return 1;
          }
     }
     return 0;

}

void removeBag (struct arrayBagStack * b, TYPE v) {

     assert(b->count > 0);
     for(int i = 0; i < b->count; i++){
          if(b->data[i] == v){
               for(int j = i; j < (b->count-1); j++){
                    b->data[j] = b→data[j+1];
               }
               b->data[b->count-1] = 0;
               b->count -= 1;
          }
     }

}

int sizeBag (struct arrayBagStack * b) {

     return b->count;

}


/* Stack Implementation */
void pushStack (struct arrayBagStack * b, TYPE v) {

     assert(b->count != MAX_SIZE);
     b->data[b->count] = v;
     b->count += 1;
```

```
}
TYPE topStack (struct arrayBagStack * b) {

     assert(b->count > 0);
     return b->data[b->count-1];

}

void popStack (struct arrayBagStack * b) {

     assert(b->count > 0);
     b->data[b->count-1] = 0;
     b->count -= 1;

int isEmptyStack (struct arrayBagStack * b) {

     if(b->count > 0){
          return 0;
     }
     else{
          return 1;
     }

}
```