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.

```
/* Bag Implementation */
void initBag (struct arrayBagStack * b){

     /* Insert your code */

}
```

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

     /* Insert your code */

}

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

     /* Insert your code */

}

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

     /* Insert your code */

}

int sizeBag (struct arrayBagStack * b) {

     /* Insert your code */


}


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

     /* Insert your code */

}
TYPE topStack (struct arrayBagStack * b) {

     /* Insert your code */

}

void popStack (struct arrayBagStack * b) {

     /* Insert your code */

}
```

```
int isEmptyStack (struct arrayBagStack * b) {

     /* Insert your code */

}
```