CS261: Exam 1

1 Problem 1: Add Bag – 40 points

Complete the C function for adding a new element to Bag implemented as a dynamic array. Input arguments of the function include the pointer to Bag and value of a new element to be added to Bag.

2 Problem 2: Remove All Bag – 60 points

Complete the C function for removing all occurrences of a given element from Bag implemented as a dynamic array. Input arguments of the function include the pointer to Bag, and the value of elements to be removed from Bag.

```
/*-----------------------------------------------*/
/* input arguments:
  da -- pointer to a dynamic array
  val -- value of the data element to be added
*/
void addDynArr(struct DynArr * da, TYPE val)
{
/*5 points; check input arguments*/
   assert(da);

/*20 points; check if there is enough capacity to add the new element*/
   if (da->size == da->capacity) _doubleCapacity(da);

/*5 points; add the new element to the end of the dynamic array*/
   da->data[da->size] = val;

/*10 points; maintain the size of the dynamic array*/
   da->size ++;
}
```

```
/*-----------------------------------------------*/
/* input arguments:
  da -- pointer to a dynamic array
  val -- value to be removed at every occurrence
*/
void removeAllDynArr(struct DynArr* da, TYPE val)
{
/*5 points; check input arguments*/
   assert(da);

   int i=0;

/*10 points; search for val*/
   while (i < da->size) {

/*10 points; check if a current element is equal to val*/
      if ( EQ(val, da->data[i]) ) /* we can also use == */
      {
/*10 points; overwrite the element that is equal to val*/
         da->data[i] = da->data[da->size-1];

/*10 points; maintain the size of the dynamic array*/
         da->size--;
      }
      else

/*15 points; correct move through the array
only when a current element is NOT equal to val */
         i++;
   }
}
```