```
/*      dynArr.h : Dynamic Array implementation. */
#ifndef DYNAMIC_ARRAY_INCLUDED
#define DYNAMIC_ARRAY_INCLUDED 1


# ifndef TYPE
# define TYPE      int
# define TYPE_SIZE sizeof(int)
# endif

# ifndef LT
# define LT(A, B) ((A) < (B))
# endif

# ifndef EQ
# define EQ(A, B) ((A) == (B))
# endif



typedef struct DynArr DynArr;

//struct DynArr
//{
//      TYPE *data;             /* pointer to the data array */
//      int size;               /* Number of elements in the array */
//      int capacity;   /* capacity ofthe array */
//};

/* Dynamic Array Functions */
DynArr *createDynArr(int cap);
void deleteDynArr(DynArr *v);

int sizeDynArr(DynArr *v);

void addDynArr(DynArr *v, TYPE val);
TYPE getDynArr(DynArr *v, int pos);
void putDynArr(DynArr *v, int pos, TYPE val);
void swapDynArr(DynArr *v, int i, int  j);
void removeAtDynArr(DynArr *v, int idx);


#endif
```