

```

#ifndef CIRCULARARRAYLIST_H_
#define CIRCULARARRAYLIST_H_

#include "list.h"

/**
 * An CircularArrayList implements the pure virtual List interface.
 *
 * It stores a list as a circular array of items, with the first item
 * in the list being in array position headPos, the second item in
 * position (headPos+1)%capacity, the third item in position
 * (headPos+2)%capacity, and so on.
 */
template <typename T>
class CircularArrayList : public List<T> {
private:
    int headPos;    // Array position of the first item in the list.
    int size;       // Number of items currently stored in this list.
    int capacity;   // Current size (including empty slots) of our array.
    T* values;      // The array that stores the items in the list.

public:
    CircularArrayList();
    ~CircularArrayList();

    int getSize();           // Get number of items in this list.
    bool isEmpty();          // True iff list contains no items.
    T peekHead();            // Returns item at front of list.
    T peekTail();            // Returns item at back of list.
    T get(int i);            // Returns the ith item in the list.

    void insertAtHead(T value); // Prepends item to front of list.
    void insertAtTail(T value); // Appends item to back of list.
    T removeHead();           // Removes and returns front item.
    T removeTail();           // Removes and returns back item.

private:
    void expandCapacity();    // Expands the array to store more items.
};

#include "circularArrayList-inl.h"

#endif // CIRCULARARRAYLIST_H_

```