```cpp
#ifndef LIST_H_
#define LIST_H_

/**
 * A List is an abstract (a.k.a. pure virtual) class specifying
 * the interface for a list of an arbitrary data type.
 */
template <typename T>
class List {
  public:
    virtual ~List() { /* do nothing */ };
    virtual int getSize() = 0;               // Get number of items in list.
    virtual bool isEmpty() = 0;              // True iff list contains no items.
    virtual T peekHead() = 0;                // Returns item at front of list.
    virtual T peekTail() = 0;                // Returns item at back of list.
    virtual T get(int i) = 0;                // Returns item in ith position.

    virtual void insertAtHead(T value) = 0;  // Prepends item to front of list.
    virtual void insertAtTail(T value) = 0;  // Appends item to back of list.
    virtual T removeHead() = 0;              // Removes and returns front item.
    virtual T removeTail() = 0;              // Removes and returns back item.
};

#endif  // LIST_H_
```