**Implement array-based list and its operators as the following description:**

|  |  |  |
| --- | --- | --- |
| **Operator** |  | **Description** |
| void makeNull(List \*L) |  | Initialize an empty list |
| int len(List L) |  | Number of elements in L |
| int empty(List L) |  | Check whether the list is empty? (0,1) |
| int fullList(List L) |  | Check whether the list is full? (0,1) |
| void print(List L) |  | Traverse the list to print out all elements |
| ElementType getAt(Position p, List L) |  | Return the element at position p |
| Position first(List L) |  | Return the first position of L |
| Position endList(List L) |  | Return the end position of L |
| void setAt(Position p, ElementType x,List \*L) |  | Update the element at position p by a new value x |
| void insertAt(Position p, ElementType x, List \*L) |  | Insert x at position p |
| ElementType popAt(Position p, List \*L) |  | Remove and return the element at position p |
| void insertFirst(ElementType x, List \*L) |  | Insert x to the first position |
| ElementType popFirst(List \*L) |  | Remove and return the first element |
| void append(ElementType x,List \*L) |  | Append a new element to the list |
| ElementType popLast(List \*L) |  | Remove and return the last element |
| Position locate(ElementType x, List L) |  | Return the position of the first appearance of x in the list |
| Position next(Position p,List L) |  | Return the next position of p in L |
| Position previous(Position p,List L) |  | Return the previous position of p in L |