

```

1 public class MyArray {
2 private int [] head = null;
3 // 4 pts
4 // add at the tail
5 public void add(int item){
6     int [] tmp ;
7     if(head == null || head.length==0)
8         tmp = new int[1];
9     else
10        tmp = new int[head.length+1];
11    for(int i = 0; i<tmp.length-1; i++)
12        tmp[i] = head[i];
13    tmp [tmp.length-1] = item;
14    head = tmp;
15 }
16 // 3 + 2 + 2 + 5 + 2 +1 = 16 pts
17 public String toString(){
18     if (head == null || head.length==0)
19         return "null";
20     String ret="["+head[0];
21     for(int i = 1; i<head.length; i++)
22         ret += ","+head[i];
23     return ret+"]";
24 }
25 // 3+ 2 + 1 + 5 + 1 = 12 pts
26
27 // add at the head
28 public void addFirst(int item){
29     int [] tmp ;
30     if(head == null || head.length==0)
31         tmp = new int[1];
32     else
33         tmp = new int[head.length+1];
34     for(int i = 0; i<tmp.length-1; i++)
35         tmp[i+1] = head[i];
36     tmp [0] = item;
37     head = tmp;
38 }
39 // 3+ 2 + 2 + 2 + 5 + 1 = 15 pts
40
41 // remove the head
42 public void remove(){
43     int [] tmp ;
44     if(head == null || head.length==0)
45         return;
46     else if (head.length==1){
47         head = null;
48         return;
49     }
50     else
51         tmp = new int[head.length-1];
52     for(int i = 0; i<tmp.length; i++)
53         tmp[i] = head[i+1];
54     head = tmp;
55 }
56 // 3 + 2 + 2 + 2 + 2 + 5 + 1 = 17 pts
57
58 // return the ith record
59 public String element(int i) {
60     if (head==null) return null;
61     if(i<0 || i>=head.length)
62         return null;
63     return ""+head[i];
64 }
65 // 3 + 2 + 2 + 2 = 9 pts
66
67 // test materials

```

```

68 // 2) add(index, object)
69 public boolean add(int index, int item){
70     int [] tmp;
71     if(head == null || head.length==0){
72         if(index!=0) return false;
73         tmp = new int[1];
74         tmp[0] = item;
75         head = tmp;
76         return true;
77     }
78     if(index < 0 || index>head.length) return false;
79     tmp = new int[head.length+1];
80     for(int i = 0; i<index; i++)
81         tmp[i] = head[i];
82     tmp[index] = item;
83     for(int i = index+1; i<tmp.length; i++)
84         tmp[i] = head[i-1];
85     head = tmp;
86     return true;
87 }
88 // 3 + 4 + 2 + 5 + 1 + 5 + 1 = 21 pts
89
90 // 3) remove (object)
91 public void remove (int item){
92     int size = 0;
93     for(int i = 0; i<head.length; i++)
94         if(head[i]!=item) size++;
95     int [] tmp = new int [size];
96     size = 0;
97     for(int i = 0; i<head.length; i++){
98         if(head[i]!=item) {
99             tmp[size++]=head[i];
100         }
101     }
102     head = tmp;
103 }
104 // 3 + 2 + 5 + 1 + 5 + 1 = 17 pts
105
106 // 5) set(index, object)
107 public void set (int index, int item){
108     if(head==null || head.length==0) return;
109     if(index<0||index>=head.length) return;
110     head[index] = item;
111 }
112 // 3 + 2 + 2 + 1 = 8 pts
113
114 // 6) int indexOf(object)
115 public int indexOf (int item){
116     if(head==null || head.length==0) return -1;
117     for(int i = 0; i<head.length; i++)
118         if(head[i]==item) return i;
119     return -1;
120 }
121 // 3 + 2 + 5 + 1 = 11 pts
122
123 // 7) object get(index)
124 public String get(int index){
125     return element(index);
126 }
127 // 5 pts
128
129 // 8) int size
130 public int size(){
131     if(head==null || head.length==0)
132         return -1;
133     else
134         return head.length;

```

```
135 }
136 // 3 + 2 + 2 = 7 pts;
137
138 // 9) boolean contains(object)
139 public boolean contains(int item){
140     if(indexOf(item)<0) return false;
141     else return true;
142 }
143 // 7 pts
144
145 // 10) clear
146 public void clear(){
147     head = null;
148 }
149 // 5 pts
150 }
```