```
package hw2;

// Matt Hyatt ... comp272

import java.util.*;

public class HW2{

// problem 3
public String replaceChar(String p, int k, char c) {
    if(k<0 || k>p.length()){
        throw new IndexOutOfBoundsException("k out of bounds");
    }
    String out = p.substring(0,k).concat(String.valueOf(c)).concat(p.substring(k+1));
    return out;
    }
// problem 4
public void timeComplexity(){
    System.out.println("i");
    System.out.println("O( n*sqrt(n) )");
    System.out.println("ii");
    System.out.println("O(n3)");
    System.out.println("O( (log(x))^10 )");

}

// problem 5
public void rowSums(int[][] arr){
    //prints the sum of the values in each row, comma separated on a single line.
    int n = arr.length;
    for(int i=0; i<n; i++){
        int sum = 0;
        for(int j=0; j<n; j++)
            sum += arr[i][j];
        System.out.print(sum);
        if(i != n-1)
            System.out.print(", ");
    }
    System.out.println();
}
public void columnMins(int[][] arr){
    int n = arr.length;
    for(int i=0; i<n; i++){
        int min = arr[i][0];
        for(int j=1; j<n; j++)
            if(arr[i][j] < min)
                min = arr[i][j];
        System.out.print(min);
```

```
        if(i != n-1)
            System.out.print(", ");
    }
    System.out.println();
}

// problem 6
public void prefixSums(LinkedList<Integer> l){
    int sum = 0;
    int i = 0;
    int size = l.size();
    for(Integer num: l){
        sum += num;
        System.out.print(sum);
        if (i != size-1)
            System.out.print(", ");
        i++;
    }
    System.out.println();
}
// problem 7
public void prefixSumsReverse(LinkedList<Integer> l){
    int sum = 0;
    int i = 0;
    int size = l.size();
    Iterator<Integer> iter = l.descendingIterator();

    while(iter.hasNext()){
        sum += iter.next();
        System.out.print(sum);
        if (i != size-1)
            System.out.print(", ");
        i++;
    }
    System.out.println();

}

// problem 8
public LinkedList<String> alphabeticConcat(LinkedList<String> a, LinkedList<String> b){
    a.addLast(null);
    b.addLast(null);

    Iterator<String> aiter = a.iterator();
    Iterator<String> biter = b.iterator();
```

```
    LinkedList<String> out = new LinkedList<>();

    String anext = aiter.next();
    String bnext = biter.next();
    while(aiter.hasNext() || biter.hasNext()){

        if (anext == null || bnext == null){
            if (anext == null) {
                out.add(bnext);
                bnext = biter.next();
            }
            if (bnext == null){
                out.add(anext);
                anext = aiter.next();
            }
        }

        else {
            if (anext.compareTo(bnext) < 0){
                out.add(anext);
                anext = aiter.next();
            }
            if (anext.compareTo(bnext) > 0){
                out.add(bnext);
                bnext = biter.next();
            }
            if (anext.compareTo(bnext) == 0){
                out.add(anext);
                out.add(bnext);
                anext = aiter.next();
                bnext = biter.next();
            }
        }

    }

    a.removeLast();
    b.removeLast();

    return out;
}

// problem 9
public ArrayList<int[]> differPairs(int[] arr, int k){
    ArrayList<int[]> out = new ArrayList<>();
```

```
    for(int item: arr){
        for(int item2: arr){
            if (item-item2 == k){
                int[] temp = new int[] {item, item2};
                out.add(temp);
            }
        }
    }
    return out;
}

public static void main(String[] args){
    String[] a = new String[]{"aa","bb","cc","dd","ee","ff","gg","hh","ii"};
    ExtLinkedList<String> l = new ExtLinkedList<>();

    for(String item: a){
        l.add(item);
    }

    System.out.println();
    System.out.println(l);
    System.out.println();

    System.out.println(1);
    System.out.println(l.secondHalfList());
    System.out.println(2);
    System.out.println(l.evenList());
    System.out.println(l.oddList());

    System.out.println(3);
    HW2 h = new HW2();
    System.out.println(h.replaceChar("boot",2,'a'));

    System.out.println(4);
    h.timeComplexity();

    System.out.println(5);
    int[][] arr = {{3,2,5},{1,0,4},{5,6,7}};
    h.rowSums(arr);
    h.columnMins(arr);

    System.out.println(6);
    LinkedList<Integer> l2 = new LinkedList<>();
    Integer[] b = {5,3,2,9,3,15,22};
    for (Integer item: b){
        l2.add(item);
```

```
    }
    h.prefixSums(l2);

    System.out.println(7);
    h.prefixSumsReverse(l2);

    System.out.println(8);
    LinkedList<String> l3 = l.oddList();
    l.add("xx");
    l.add("zz");
    System.out.println(h.alphabeticConcat(l,l3));

    System.out.println(9);
    int[] nums = {1,4,9,12, 6, 15, 5, 13,17};
    ArrayList<int[]> pairs = h.differPairs(nums,4);
    for (int[] item: pairs){
        System.out.print("[" + item[0] + "," + item[1] + "], ");
    }
    System.out.println();
}
```

public class ExtLinkedList extends LinkedList{

```
// problem 1
public ExtLinkedList <E> secondHalfList(){
    //return the back half rounded down
    int half = (size() - (size()%2))/2;
    ExtLinkedList<E> out = new ExtLinkedList();

    Iterator<E> iter = this.iterator();

    int i = size()-1;
    while(iter.hasNext()){
        E item = iter.next();
        if(i<half){
            out.add(item);
        }
        i--;
    }

    return out;
}

//problem 2
public ExtLinkedList <E> oddList(){
    Iterator<E> iter = this.iterator();
    ExtLinkedList<E> out = new ExtLinkedList();
```

```
    int i = 0;

    while(iter.hasNext()){
        E item = iter.next();
        if (i%2 == 1){
            out.add(item);
        }
        i++;
    }
    return out;
}
public ExtLinkedList<E> evenList(){
    Iterator<E> iter = this.iterator();
    ExtLinkedList<E> out = new ExtLinkedList();
    int i = 0;

    while(iter.hasNext()){
        E item = iter.next();
        if (i%2 == 0){
            out.add(item);
        }
        i++;
    }
    return out;

}
}
```