Project6\_1:

**import** java.util.\*;

**public** **class** Project6\_1 {

**public** **static** **void** main(String[] args){

Scanner read = **new** Scanner(System.***in***);

String phrase = **new** String();

linkedList mylist = **new** linkedList();

**char** cont = ' ';

**do**{

**boolean** plaindrome = **true**;

System.***out***.print("Enter a phrase:");

phrase = read.nextLine();

**for**(**int** i = 0; i < phrase.length(); i++){

**if**(Character.*isAlphabetic*(phrase.charAt(i))){

**char** temp = Character.*toLowerCase*(phrase.charAt(i));

mylist.insertStack(temp);

mylist.insertQueue(temp);

}

}

//mylist.display(mylist.queue);

//mylist.display(mylist.stack);

**while**(mylist.stack!= **null** && mylist.queue != **null** && plaindrome ){

**if**(mylist.stack.info == mylist.queue.info){

//System.out.println("Queue: " + mylist.queue.info + " Stack: " + mylist.stack.info);

mylist.deleteStack();

mylist.deleteQueue();

}**else**{

plaindrome = **false**;

}

}

**if**(plaindrome){

System.***out***.println("Is a PALINDROME");

}**else**{

System.***out***.println("Not a PLAINDROME");

}

mylist.stack = **null**;

mylist.queue = **null**;

System.***out***.print("Continue(y/n)?");

cont = read.next().charAt(0);

String temp = read.nextLine();

}**while**(cont == 'y' || cont == 'Y');

}

}

output:

Enter a phrase:a toyota

Is a PALINDROME

Continue(y/n)?y

Enter a phrase:afsfa

Is a PALINDROME

Continue(y/n)?y

Enter a phrase:fdk

Not a PLAINDROME

Continue(y/n)?

Project6\_2:

**import** java.util.Scanner;

**public** **class** Project6\_2 {

**public** **static** **void** main(String[] args){

linkedList mylist = **new** linkedList();

Scanner read = **new** Scanner(System.***in***);

**int** termA, termB;

**boolean** first = **true**;

**do**{

System.***out***.print("Enter the coefficient and the power of a term(0 0) to stop: ");

termA=read.nextInt(); termB=read.nextInt();

mylist.insertQueue(termA);

mylist.insertQueue(termB);

String temp = read.nextLine();

}**while**(termA != 0 && termB != 0);

System.***out***.print("F(X) = ");

node coeff = mylist.queue;

node pow = coeff.next;

**while**(pow.next != **null**){

//mylist.deleteQueue();

**if**(coeff.info != 0 && pow.info != 0){

**if**(!first){

System.***out***.print(" + ");

}

first = **false**;

System.***out***.printf("%dX^%d", coeff.info, pow.info);

coeff = coeff.next.next;

pow = coeff.next;

}

}

System.***out***.println();

System.***out***.printf("Enter the X value: ");

**int** x = read.nextInt();

**int** result = 0;

coeff = mylist.queue;

pow = coeff.next;

**while**(pow.next != **null**){

//mylist.deleteQueue();

**if**(coeff.info != 0 && pow.info != 0){

result += coeff.info \* Math.*pow*(x,pow.info);

coeff = coeff.next.next;

pow = coeff.next;

}

}

System.***out***.printf("F(%d) = %d", x, result);

}

}

output:

Enter the coefficient and the power of a term(0 0) to stop: 3 2

Enter the coefficient and the power of a term(0 0) to stop: 5 1

Enter the coefficient and the power of a term(0 0) to stop: 0 0

F(X) = 3X^2 + 5X^1

Enter the X value: 4

F(4) = 68

Project6\_3:  
import java.util.\*;

import java.io.\*;

public class Project6\_3 {

public static void main(String[] args){

String[] queue = new String[5];

linkedList waitinglist = new linkedList();

try {

copydata("waiting\_sheet", waitinglist);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("The queue before the Chair calls student one-by-one");

display(waitinglist);

while(waitinglist.squeue!=null){

studentNode min = findMin(waitinglist);

System.out.println(min.name + " is served, now the queue is");

display(waitinglist);

}

}

// change to squeue

public static void copydata(String fName, linkedList waitinglist) throws Exception{

BufferedReader in = new BufferedReader(new FileReader(fName));

String line;

int i = 0;

while((line = in.readLine()) != null){

String token[] = line.split(" ", 2);

waitinglist.insertQueue(token[0],Integer.parseInt(token[1]));

i++;

};

in.close();

}

public static studentNode findMin(linkedList waitinglist){

studentNode min = waitinglist.squeue;

studentNode index = waitinglist.squeue.next;

studentNode minParent = waitinglist.squeue;

studentNode indexParent = waitinglist.squeue;

if(index == null){

//System.out.println("end");

waitinglist.squeue = min.next;

min.next = null;

return min;

}

while(index!=null){

if(index.miniutes < min.miniutes){

min = index;

minParent = indexParent;

index = index.next;

indexParent = indexParent.next;

}else{

index=index.next;

indexParent = indexParent.next;

}

}

//System.out.print("###");

if(min == waitinglist.squeue){

waitinglist.squeue = min.next;

min.next = null;

}else{

minParent.next = min.next;

//System.out.print(minParent.name);

min.next = null;

}

return min;

}

public static void display(linkedList waitinglist){

System.out.printf("squeue");

String name = new String();

studentNode p = waitinglist.squeue;

while(p!=null){

name = p.name;

System.out.print(" --> " + name + "," + p.miniutes);

p=p.next;

}

System.out.println(" --> NULL");

}

}

output:

The queue before the Chair calls student one-by-one

squeue --> Bob,10 --> Mary,7 --> Bill,5 --> Tom,15 --> Jack,3 --> NULL

Jack is served, now the queue is

squeue --> Bob,10 --> Mary,7 --> Bill,5 --> Tom,15 --> NULL

Bill is served, now the queue is

squeue --> Bob,10 --> Mary,7 --> Tom,15 --> NULL

Mary is served, now the queue is

squeue --> Bob,10 --> Tom,15 --> NULL

Bob is served, now the queue is

squeue --> Tom,15 --> NULL

Tom is served, now the queue is

squeue --> NULL