Day8\_Assignment:

1. package day8\_Assignment;

/\* Add Elements at First and Last Position

Write a program to:

•Create a LinkedList of integers.

•Add elements at the beginning and at the end.

•Display the updated list.\*/

import java.util.LinkedList;

import java.util.List;

public class Pro1 {

public static void main(String[] args) {

LinkedList<Integer> list=new LinkedList<Integer>(List.*of*(1,2,3,4,5));

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

list.add(0,20);

list.add(list.size(),30);

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

}

}

Output:

[1, 2, 3, 4, 5]

[20, 1, 2, 3, 4, 5, 30]

2. package day8\_Assignment;

import java.util.Iterator;

import java.util.LinkedList;

import java.util.List;

/\*Search for an Element

Write a program to:

•Create a LinkedList of Strings.

•Ask the user for a string to search.

•Display if the string is found or not.\*/

public class Pro2 {

public static void main(String[] args) {

LinkedList<String> list=new LinkedList<String>(List.*of*("Hi","Hello","bye"));

Iterator<String> itr=list.iterator();

while(itr.hasNext()) {

if(itr.next().equals("bye")) {

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

}

}

}

}

Output:

Found

3. package day8\_Assignment;

import java.util.LinkedList;

import java.util.List;

import java.util.ListIterator;

/\*Write a program to:

•Create a LinkedList of cities.

Use ListIterator to display the list in both forward and reverse directions.\*/

public class Pro3 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

LinkedList<String> citys=new LinkedList<String>(List.*of*("kdp","kmp","ap"));

ListIterator<String> itr=citys.listIterator();

System.***out***.println("Citys in forward direction");

while(itr.hasNext()) {

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

}

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

System.***out***.println("Citys in backward direction");

while(itr.hasPrevious()) {

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

}

//System.out.println(citys.reversed());

}

}

Output:

Citys in forward direction

kdp kmp ap

Citys in backward direction

ap kmp kdp

4. package day8\_Assignment;

import java.util.ArrayList;

import java.util.LinkedList;

import java.util.List;

/\*8. Convert LinkedList to ArrayList

Write a program to:

•Create a LinkedList of Strings.

•Convert it into an ArrayList.

•Display both the LinkedList and ArrayList.\*/

public class Pro4 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

LinkedList<String> citys=new LinkedList<String>(List.*of*("kdp","kmp","ap"));

ArrayList<String> citys2=new ArrayList<String>(citys);

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

System.***out***.println(citys2);

Object c=citys.clone();

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

}

}

Output:

[kdp, kmp, ap]

[kdp, kmp, ap]

[kdp, kmp, ap]

5. package day8\_Assignment;

import java.util.ArrayList;

import java.util.Scanner;

/\*Search an Element

Write a program to:

•Create an ArrayList of integers.

•Ask the user to enter a number.

•Check if the number exists in the list.\*/

public class Que1 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

System.***out***.println("enter num to check: ");

int n=sc.nextInt();

boolean exist=false;

ArrayList<Integer> list=new ArrayList<Integer>();

list.add(1);

list.add(2);

list.add(3);

for(Integer i:list) {

if(i==n) {

exist=true;

break;

}

}

if(exist==true) {

System.***out***.println("num exists");

}

else {

System.***out***.println("num doesn't exists");

}

}

}

Output:

enter num to check:

9

num doesn't exists

6. package day8\_Assignment;

import java.util.ArrayList;

import java.util.List;

/\*Remove Specific Element

Write a program to:

•Create an ArrayList of Strings.

•Add 5 fruits.

•Remove a specific fruit by name.

•Display the updated list.\*/

public class Que2 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList<String> list=new ArrayList<String>(List.*of*("apple","banana","orange","kiwi","mango"));

for(String s:list) {

if(s=="kiwi") {

list.remove(s);

}

}

System.***out***.println("Updated list: "+list);

}

}

Output:

Updated list: [apple, banana, orange, mango]

7. package day8\_Assignment;

import java.util.ArrayList;

import java.util.List;

/\*Sort Elements

Write a program to:

•Create an ArrayList of integers.

•Add at least 7 random numbers.

•Sort the list in ascending order.

•Display the sorted list.\*/

public class Que3 {

public static void main(String[] args) {

ArrayList<Integer> list=new ArrayList<Integer>(List.*of*(1,3,12,45,67,89,71));

for(int i=0;i<list.size()-1;i++) {

int n1=list.get(i);

int n2=list.get(i+1);

if(n1>n2) {

n1=n1+n2;

n2=n1-n2;

n1=n1-n2;

}

}

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

}

}

Output:

[1, 3, 12, 45, 67, 89, 71]

8. package day8\_Assignment;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

/\* Reverse the ArrayList

Write a program to:

•Create an ArrayList of characters.

•Add 5 characters.

•Reverse the list using Collections.reverse() and display it.\*/

public class Que4 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList<Character> list=new ArrayList<Character>(List.*of*('a','c','v','b','e'));

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

Collections.*reverse*(list);

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

}

}

Output:

[a, c, v, b, e]

[e, b, v, c, a]

9. package day8\_Assignment;

import java.util.ArrayList;

/\*Update an Element

Write a program to:

•Create an ArrayList of subjects.

•Replace one of the subjects (e.g., “Math” to “Statistics”).

•Print the list before and after the update.\*/

public class Que5 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList<String> subjects=new ArrayList<String>();

subjects.add("Maths");

subjects.add("Physics");

subjects.add("chemistry");

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

subjects.set(0,"Mathematics");

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

}

}

Output:

[Maths, Physics, chemistry]

[Mathematics, Physics, chemistry]

10. package day8\_Assignment;

import java.util.ArrayList;

import java.util.List;

/\* Remove All Elements

Write a program to:

•Create an ArrayList of integers.

•Add multiple elements.

•Remove all elements using clear() method.

•Display the size of the list.\*/

public class Que6 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList<Integer> list=new ArrayList<Integer>(List.*of*(1,2,3,4,5));

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

list.clear();

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

}

}

Output:

[1, 2, 3, 4, 5]

0

11. package day8\_Assignment;

import java.util.ArrayList;

import java.util.List;

/\*Write a program to:

•Create an ArrayList with some elements.

•Create a second ArrayList.

•Copy all elements from the first to the second using addAll() method.\*/

public class Que7 {

public static void main(String[] args) {

ArrayList<Integer> l1=new ArrayList<Integer>(List.*of*(1,2,3,4,5));

ArrayList<Integer> l2=new ArrayList<Integer>();

l2.addAll(l1);

System.***out***.println(l2);

}

}

Output:

[1, 2, 3, 4, 5]

12. package day8\_Assignment;

import java.util.ArrayList;

import java.util.Iterator;

/\* Store Custom Objects

Write a program to:

•Create a class Student with fields: id, name, and marks.

•Create an ArrayList of Student objects.

•Add at least 3 students.

•Display the details using a loop.\*/

public class Student {

int id;

String name;

double marks;

Student(int id,String name,double marks){

this.id=id;

this.name=name;

this.marks=marks;

}

public static void main(String[] args) {

ArrayList<Student> stu=new ArrayList<Student>();

Student s1=new Student(1,"Spandana",90.0);

Student s2=new Student(1,"Swapna",85.0);

Student s3=new Student(3,"Jyoshnana",91.0);

stu.add(s1);

stu.add(s2);

stu.add(s3);

Iterator<Student> itr=stu.iterator();

while(itr.hasNext()) {

Student detail=itr.next();

System.***out***.println(detail.id+" "+detail.name+" "+detail.marks);

}

}

}

Output:

1 Spandana 90.0

1 Swapna 85.0

3 Jyoshnana 91.0

13. import java.util.\*;

public class AllInOneStack {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Stack<Integer> intStack = new Stack<>();

for (int i = 1; i <= 5; i++) intStack.push(i);

System.out.println(intStack);

System.out.println(intStack.pop());

System.out.println(intStack.peek());

System.out.println(intStack.isEmpty());

String input = sc.nextLine();

Stack<Character> charStack = new Stack<>();

for (char c : input.toCharArray()) charStack.push(c);

StringBuilder rev = new StringBuilder();

while (!charStack.isEmpty()) rev.append(charStack.pop());

System.out.println(rev);

String expr = sc.nextLine();

System.out.println(isBalanced(expr) ? "Valid" : "Invalid");

int num = sc.nextInt();

System.out.println(decimalToBinary(num));

sc.close();

}

static boolean isBalanced(String s) {

Stack<Character> stack = new Stack<>();

for (char ch : s.toCharArray()) {

if (ch == '(' || ch == '{' || ch == '[') stack.push(ch);

else if (ch == ')' && (stack.isEmpty() || stack.pop() != '(')) return false;

else if (ch == '}' && (stack.isEmpty() || stack.pop() != '{')) return false;

else if (ch == ']' && (stack.isEmpty() || stack.pop() != '[')) return false;

}

return stack.isEmpty();

}

static String decimalToBinary(int n) {

if (n == 0) return "0";

Stack<Integer> stack = new Stack<>();

while (n > 0) {

stack.push(n % 2);

n /= 2;

}

StringBuilder bin = new StringBuilder();

while (!stack.isEmpty()) bin.append(stack.pop());

return bin.toString();

}

}

Output:

[1, 2, 3, 4, 5]

5

4

False

14.import java.util.Deque;

import java.util.LinkedList;

import java.util.Scanner;

public class PalindromeCheck {

public static boolean isPalindrome(String s) {

Deque<Character> dq = new LinkedList<>();

for (char c : s.toCharArray()) dq.add(c);

while (dq.size() > 1) {

if (!dq.pollFirst().equals(dq.pollLast())) return false;

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String input = sc.nextLine();

sc.close();

System.out.println(isPalindrome(input) ? "Palindrome" : "Not a palindrome");

}

}

Output:

madam

Palindrome