**Bright Students Assignment  
Pranay Kuhite  
CSAIML-A  
Roll no-48**

# OOPS Assignment

## **1)Problem Statement: Linear Search**

\*Caroline is from Chicago. She is looking forward to her 10 days trip to Switzerland. She has decided to make Zurich as her base city and visit one city each day from Zurich by doing day trips. She has a list of 10 cities and towns in Switzerland that she wants to cover but has to remove 2 out of the list as on the 10th day she has a flight back to Chicago. She has decided to sort the list based on the distance in kms of each city from Zurich.\*

\*Task: Use linear search to perform this task.\*

- Find the city which is 52 kms from Zurich

- Find the city which is greater than or equal to 270 kms from Zurich.

\*\*Note : Use a linear search algorithm to display the number of comparisons.\*\*

Given below is the list of places along with their distance.

- Bern: 138 kms

- Lucerne: 52 kms

- Interlaken: 118 Kms

- Grindelwald: 136 kms

- Engelberg: 85 kms

- Geneva: 276 kms

- Murren: 103 kms

- Basel: 87 kms

- Zermatt: 214 kms

- Jungfraujoch: 101 kms

\*Sample Input:\*

String[] citiesFromZurich ={"Bern","Lucerne","InterLaken","Grindelwald","Engelberg","Geneva","Murren","Basel","Zermatt","Jungfraujoch"};

int[] distanceFromZurich = {138,52,118,136,85,276,103,87,214,101};

\*Expected Output:\*   
City at 52 kms from Zurich : Lucerne Number of comparisons:: 10   
Cities more than 270 kms from Zurich : {Geneva}

## CODE

package trial;

public class linear\_search {

static int *comparisonsCount* = 0;

static String findCityAtDistance(String[] cities, int[] distances, int targetDistance) {

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

*comparisonsCount*++;

if (distances[i] == targetDistance) {

return cities[i];

}

}

return "Not Found";

}

static String[] findCitiesMoreThanDistance(String[] cities, int[] distances, int targetDistance) {

int count = 0;

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

*comparisonsCount*++;

if (distances[i] >= targetDistance) {

count++;

}

}

String[] result = new String[count];

count = 0;

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

if (distances[i] >= targetDistance) {

result[count] = cities[i];

count++;

}

}

return result;

}

public static void main(String[] args) {

String[] citiesFromZurich = {"Bern", "Lucerne", "InterLaken", "Grindelwald", "Engelberg", "Geneva", "Murren", "Basel", "Zermatt", "Jungfraujoch"};

int[] distanceFromZurich = {138, 52, 118, 136, 85, 276, 103, 87, 214, 101};

String cityAt52Kms = *findCityAtDistance*(citiesFromZurich, distanceFromZurich, 52);

System.***out***.println("City at 52 kms from Zurich: " + cityAt52Kms);

System.***out***.println("Number of comparisons: " + *comparisonsCount*);

*comparisonsCount*=0;

String[] citiesMoreThan270Kms = *findCitiesMoreThanDistance*(citiesFromZurich, distanceFromZurich, 270);

System.***out***.print("Cities more than 270 kms from Zurich: ");

for (String city : citiesMoreThan270Kms) {

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

}

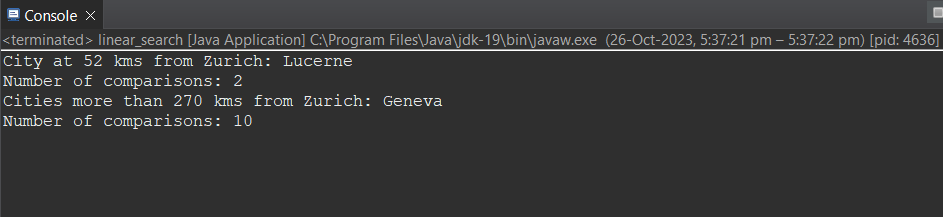
System.***out***.println();

System.***out***.println("Number of comparisons: " + *comparisonsCount*);

}

}

## OUTPUT



## **7) # Problem Statement: Palindrome**

\*Palindrome is a word, sentence or a number that reads the same even after its order is reversed. Harrah is one such palindrome town in Oklahoma, United States. For administrative purposes, a list of all palindrome towns and cities is being prepared.\*

\*Write a Java program that states whether the name of a place is a palindrome or not. Print the words and display their total count.\*

\*Sample Input\*   
String words []= {"Hannah", "Texas", "Renner", "Eleele" }

\*Expected Output\*   
Hannah is a palindrome  
Renner is a palindrome  
Number of Palindromes: 2

## CODE

package trial;

public class palindrome {

static boolean isPalindrome(String word) {

word = word.toLowerCase();

int wordstart = 0;

int wordend = word.length() - 1;

while (wordstart < wordend) {

if (word.charAt(wordstart) != word.charAt(wordend)) {

return false;

}

wordstart++;

wordend--;

}

return true;

}

public static void main(String[] args) {

String words[] = {"Hannah", "Texas", "Renner", "Eleele"};

int palindromeCount = 0;

for (String word : words) {

if (*isPalindrome*(word)) {

System.***out***.println(word + " is a palindrome");

palindromeCount++;

}

}

System.***out***.println("Total number of palindromes: " + palindromeCount);

}

}

## OUTPUT

