Problem 1:

This program will initialize an array with ten random integers and then print out the following:

every element at an even index,

every even element,

all elements in reverse order,

only the first and last elements,

the minimum and maximum element,

the sum of all elements and

the alternating sum of all elements(the alternating sum contains all elements at even index added, and the elements at odd index subtracted).

begin:

import Random

class ArrayQuestion{

initiate int max = Integer.MIN\_VALUE, min = Integer.MAX\_VALUE, sum = 0, evenSum = 0, oddSum = 0

Random rand = new Random();

int[] intArray = new int[10];

function createArray(){

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

assign a random integer to intArray at index i

}

display "The array created: "

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

display intArray[i] + " "

}

display "The elements at the even index: "

for (int i = 0; i < intArray.length; i = i+2) {

display intArray[i]

}

display "The even elements are: "

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

if (intArray [i] >= max){

max = intArray[i]

}

if (intArray[i] <= min){

min = intArray[i]

}

if (intArray[i] % 2 == 0){

display intArray[i] + " "

}

}

display "The array in reverse is: "

for (int i = intArray.length-1; i > 0; i--) {

if(i % 2 != 0){

oddSum = oddSum + intArray[i]

}

display intArray[i] + " "

sum = sum + intArray[i]

}

display "The first element of the array is " + intArray[0]

display "The last element of the array is " + intArray[intArray.length - 1]

display "The max element in the array is " + max

display "The min element in the array is " + min

display "The sum of all elements is " + sum

display "The alternating sum is " + (evenSum - oddSum)

}

}

end

Problem 2

This program reads a sequence of input values and displays a bar chart of the values in data, using asterisks. That max value's bar would be drawn with 40 asterisks. Shorter values will be shown with proportionally fewer asterisks.

begin:

import Scanner

class AsteriksGraph{

initiate int values, numberOfInputs;

initiate double max = Double.MIN\_VALUE;

initiate double[] doubleArr;

initiate String[] strArray, astriexArray;

Scanner keyboard = new Scanner(System.in);

function inputData(){

display "Welcome, how many values will you be importing: "

accept input into numberOfInputs

set doubleArr, strArray and astriexArray to .length = numberOfInputs

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

display "Enter caption: "

store caption to strArray[i]

display "Enter value: "

store value into doubleArr[i]

}

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

if(doubleArr[i] >= max){

max = doubleArr[i]

}

}

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

int astriexAmount = (int) (40 \* doubleArr[i]/max)

astriexArray[i] = "\*".repeat(asteriexAmount)

}

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

display strArray[i] + " " + astriexArray[i]

}

}

}

end

Problem 3

This program will read a sequence of up to 25 pairs of names and postal(Zip) codes for individuals.

begin

import Scanner

class StoreNameZip{

Scanner keyboard = new Scanner(System.in)

initiate private int count = 0

initiate private String storageName

initiate private int[] zipArray = new int[25]

initiate private String[] fNameArray = new String[25], lNameArray = new String[25]

function StoreNameZip(String storageName){

this.storageName = storageName

intoArray();

}

function intoArray() {

display "Welcome, we will be storing your data into " + storageName

display "Please prepare a max of 25 first names, last names and zip codes"

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

display "Enter first name(" + i + "): "

fNameArray[i] = the next input

display "Enter last name(" + i + "): "

lNameArray[i] = the next input

display "Enter zip code(" + i + "): "

zipArray[i] = the next input

display "Type '1' to exit, or '0' to continue"

int quit = next input

if(quit == 1) {

break;

}

}

System.out.println("Here is the created data: ");

printArray();

}

function printArray() {

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

display fNameArray[i] + " " + lNameArray[i] + " " + zipArray[i]

}

}

}

end

Problem 4

This program promps

A theater seating chart is implemented as a two-dimensional array of ticket prices like presented below. Write a program that prompts the users to pick either a seat or a price. Mark sold seats by changing the price to zero. When a user specifies a seat, make sure it is available. When a user specifies a price, find any seat with that price. Verify that the price is among those offered by the theater.

begin

class TheaterClass{

initiate [][] theaterSeats = { { 10, 10, 10, 10, 10, 10, 10, 10, 10, 10 }, { 10, 10, 10, 10, 10, 10, 10, 10, 10, 10 },

{ 10, 10, 10, 10, 10, 10, 10, 10, 10, 10 }, { 10, 10, 20, 20, 20, 20, 20, 20, 10, 10 },

{ 10, 10, 20, 20, 20, 20, 20, 20, 10, 10 }, { 10, 10, 20, 20, 20, 20, 20, 20, 10, 10 },

{ 20, 20, 30, 30, 40, 40, 30, 30, 20, 20 }, { 20, 30, 30, 40, 50, 50, 40, 30, 30, 20 },

{ 30, 40, 50, 50, 50, 50, 50, 50, 40, 30 } }

function printTheatsSeats() {

for (int i = 0; i < theaterSeats.length; i++) { // this equals to the row in our matrix.

for (int j = 0; j < theaterSeats[i].length; j++) { // this equals to the column in each row.

display theaterSeats[i][j] + " "

}

}

}

function returnSeat(int price) {

initiate int row = 0, column = 0;

if (price < 10 || price > 50) {

display "There are no seats with that price"

}

else {

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

for (int j = 0; j < theaterSeats[i].length; j++) {

if (theaterSeats[i][j] == price) {

row = i;

column = j;

}

}

}

display "Avalible seat at (" + row + ", " + column + ")";

theaterSeats[row][column] = 0;

}

}

function returnPrice(int row, int column) {

if (row <= 9 & column <= 10) {

display "Please pay " + theaterSeats[row + 1][column + 1] + " at the register"

theaterSeats[row][column] = 0;

} else {

display "That seats does not exist, please try again";

}

}

}

end

Problem 5

This program will return the total, average, row total, column total, and the highest and lowest in the row of a 2D array.

begin

class ArrayOperations{

function getTotal(2D Array){

int total = 0

for (int i = 0; i < x.length; i++) { // this equals to the row in our matrix.

for (int j = 0; j < x[i].length; j++) { // this equals to the column in each row.

total = total + x[i][j]

}

}

return total

}

function getAverage(2D Array){

initaite double total = 0, count = 0

for (int i = 0; i < x.length; i++) { // this equals to the row in our matrix.

for (int j = 0; j < x[i].length; j++) { // this equals to the column in each row.

total = total + x[i][j]

count++

}

}

return total / count

}

function getRowTotal(2D Array, row){

double rowTotal = 0

for(int j = 0, j < x[0].length; j++){

rowTotal = rowTotal + x[row][j]

}

return rowTotal

}

function double getHighestInRow(int[][] x, int row) {

double max = Double.MIN\_VALUE;

for (int j = 0; j < x[0].length; j++) { // this equals to the column in each row.

if (x[row][j] > max) {

max = x[row][j]

}

}

return max;

}

function getLowestInRow(int[][] x, int row) {

double min = Double.MAX\_VALUE;

for (int j = 0; j < x[0].length; j++) { // this equals to the column in each row.

if (x[row][j] > min) {

min = x[row][j]

}

}

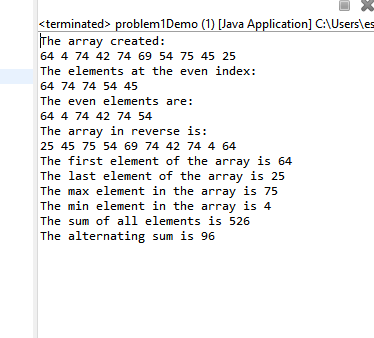
return min;

}

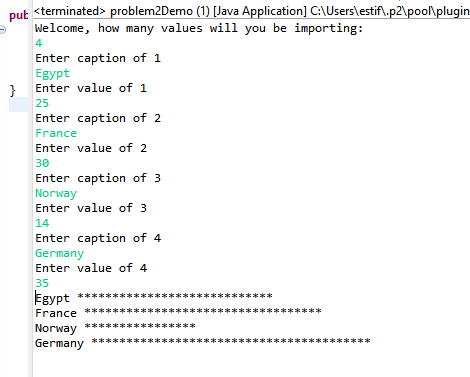
}

end

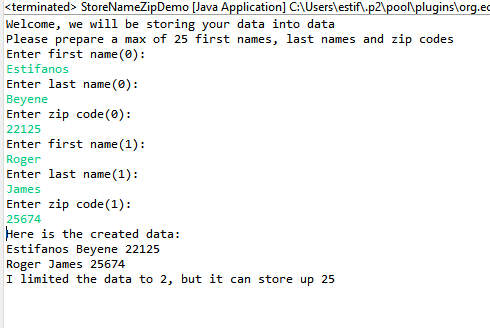
Problem 1:



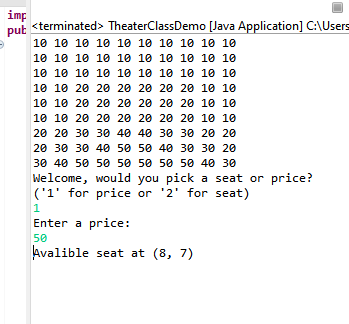
Problem 2:

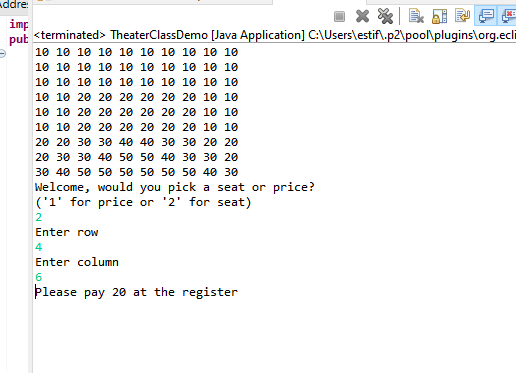


Problem 3



Problem 4





Problem 5

