**Lab Task#01:**

1.Write a program for Selection sort that sorts an array containing numbers, prints all the sort values of array each followed by its location.

**Code: Output:**

import java.util.\*;

public class DSALab05 {

public static void main(String[] args) {

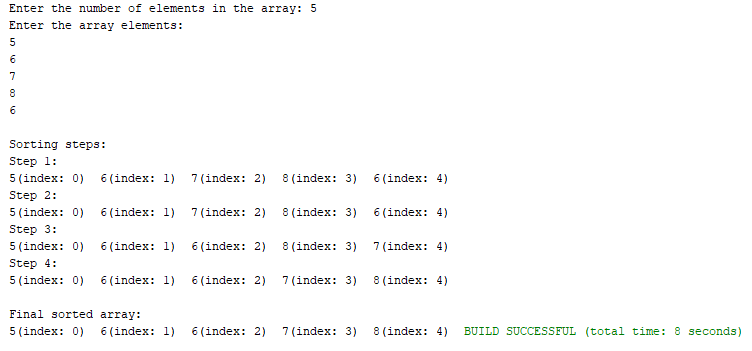
Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements in the array: ");

int n = scanner.nextInt();

int[] arr = new int[n];

System.out.println("Enter the array elements:");

 for (int i = 0; i < n; i++) {

arr[i] = scanner.nextInt(); }

System.out.println("\nSorting steps:");

for (int i = 0; i < arr.length - 1; i++) {

int minIndex = i;

for (int j = i + 1; j < arr.length; j++) {

if (arr[j] < arr[minIndex]) {

minIndex = j; }

}

int temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

System.out.println("Step " + (i + 1) + ":");

for (int k = 0; k < arr.length; k++) {

System.out.print(arr[k] + "(index: " + k + ") "); }

System.out.println();

}

System.out.println("\nFinal sorted array:");

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

System.out.print(arr[i] + "(index: " + i + ") "); }

}

}

**Lab Task#02:**

Write a program that takes 10 numbers as input in an array. Sort the elements of array by using Bubble sort. Print each iteration of the sorting process.

**Code: Output:**

import java.util.\*;

public class DSALab05 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[] arr = new int[10];

System.out.println("Enter 10 numbers:");

A screenshot of a computer

Description automatically generated for (int i = 0; i < 10; i++) {

arr[i] = scanner.nextInt(); }

for (int i = 0; i < arr.length - 1; i++) {

for (int j = 0; j < arr.length - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp; }

}

System.out.println("Iteration " + (i + 1) + ":");

for (int k = 0; k < arr.length; k++) {

System.out.print(arr[k] + " "); }

System.out.println(); }

System.out.println("Sorted array:");

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

System.out.print(arr[i] + " "); }

}

}

**Lab Task#03:**

3.Write a program that takes 10 random numbers in an array. Sort the elements of array by using Merge sort applying recursive technique. Print each iteration of the sorting process.

**Code: Output:**

import java.util.\*;

public class DSALab05 {

public static void main(String[] args) {

Random random = new Random();

int[] arr = new int[10];

System.out.println("Original array:");

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

arr[i] = random.nextInt(100);

System.out.print(arr[i] + " ");

}

System.out.println();

mergeSort(arr, 0, arr.length - 1);

System.out.println("Sorted array:");

for (int num : arr) {

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

}

}

public static void mergeSort(int[] arr, int left, int right) {

if (left < right) {

int mid = left + (right - left) / 2;

mergeSort(arr, left, mid);

mergeSort(arr, mid + 1, right);

merge(arr, left, mid, right);

System.out.print("After merging [" + left + ", " + right + "]: ");

for (int num : arr) {

System.out.print(num + " ");}

System.out.println(); }

}

public static void merge(int[] arr, int left, int mid, int right) {

int n1 = mid - left + 1;

A screenshot of a computer code

Description automatically generated int n2 = right - mid;

int[] leftArray = new int[n1];

int[] rightArray = new int[n2];

for (int i = 0; i < n1; i++) {

leftArray[i] = arr[left + i]; }

for (int j = 0; j < n2; j++) {

rightArray[j] = arr[mid + 1 + j];}

int i = 0, j = 0, k = left;

while (i < n1 && j < n2) {

if (leftArray[i] <= rightArray[j]) {

arr[k] = leftArray[i];

i++; } else {

arr[k] = rightArray[j];

j++; }

k++; }

while (i < n1) {

arr[k] = leftArray[i];

i++;

k++; }

while (j < n2) {

arr[k] = rightArray[j];

j++;

k++; }

}

}

**Home Task#01:**

1.Declare an array of size n to store account balances. Initialize with values 0 to 100000 and sort Account No’s according to highest balance values by using Quick sort.

**Code: Output:**

import java.util.\*;

public class DSALab05 {

static class Account {

int accountNo;

int balance;

Account(int accountNo, int balance) {

this.accountNo = accountNo;

this.balance = balance;}

}

public static void main(String[] args) {

Random random = new Random();

int n = 10;

Account[] accounts = new Account[n];

for (int i = 0; i < n; i++) {

int accountNo = 1000 + i;

int balance = random.nextInt(100001);

accounts[i] = new Account(accountNo, balance); }

System.out.println("Original Account Balances:");

for (Account account : accounts) {

System.out.println("Account No: " + account.accountNo + " Balance: " + account.balance);

}

quickSort(accounts, 0, n - 1);

System.out.println("\nSorted Account Balances (Highest Balance First):");

for (Account account : accounts) {

System.out.println("Account No: " + account.accountNo + " Balance: " + account.balance);}

}

public static void quickSort(Account[] accounts, int low, int high) {

if (low < high) {

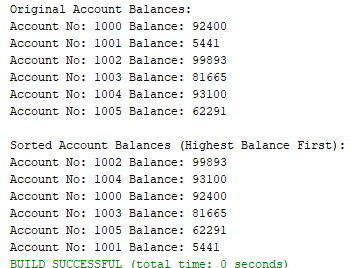
int pi = partition(accounts, low, high);

quickSort(accounts, low, pi - 1);

quickSort(accounts, pi + 1, high); }

}

public static int partition(Account[] accounts, int low, int high) {

 int pivot = accounts[high].balance;

int i = (low - 1);

for (int j = low; j < high; j++) {

if (accounts[j].balance >= pivot) {

i++;

Account temp = accounts[i];

accounts[i] = accounts[j];

accounts[j] = temp;}

}

Account temp = accounts[i + 1];

accounts[i + 1] = accounts[high];

accounts[high] = temp;

return i + 1;}

}**Home Task#02:**

2. Write a program which takes an unordered list of integers (or any other objects e.g. String), you have to rearrange the list in their natural order using merge sort

**Code:**

import java.util.\*;

public class DSALab05 {

public static <T extends Comparable<T>> void mergeSort(List<T> list) {

if (list.size() <= 1) {

return; }

int mid = list.size() / 2;

List<T> left = new ArrayList<>(list.subList(0, mid));

List<T> right = new ArrayList<>(list.subList(mid, list.size()));

mergeSort(left);

mergeSort(right);

merge(list, left, right); }

private static <T extends Comparable<T>> void merge(List<T> list, List<T> left, List<T> right) {

int i = 0, j = 0, k = 0;

while (i < left.size() && j < right.size()) {

if (left.get(i).compareTo(right.get(j)) <= 0) {

list.set(k++, left.get(i++));

} else {

list.set(k++, right.get(j++)); }

}

while (i < left.size()) {

list.set(k++, left.get(i++));}

while (j < right.size()) {

list.set(k++, right.get(j++));}}

public static void main(String[] args) {

List<Integer> integers = new ArrayList<>(Arrays.asList(34, 7, 23, 32, 5, 62, 32));

System.out.println("Original List (Integers): " + integers);

mergeSort(integers);

System.out.println("Sorted List (Integers): " + integers);

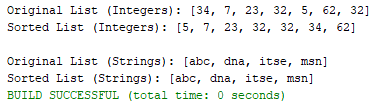
List<String> strings = new ArrayList<>(Arrays.asList("abc", "dna", "itse", "msn"));

System.out.println("\nOriginal List (Strings): " + strings);

mergeSort(strings);

System.out.println("Sorted List (Strings): " + strings);}

}**Output:**



**Home Task#03:**

3.You are given an unordered list of integers or strings. Write a program to Take this list as input. Sort it in naturalorder using Merge Sort. For integers, this means ascending order. For strings, this means alphabetical order. Print the sorted list.

**Code: Output:**

import java.util.\*;

public class DSALab05 {

public static <T extends Comparable<T>> void mergeSort(List<T> list) {

if (list.size() <= 1) return;

int mid = list.size() / 2;

List<T> left = new ArrayList<>(list.subList(0, mid));

List<T> right = new ArrayList<>(list.subList(mid, list.size()));

mergeSort(left);

mergeSort(right);

merge(list, left, right); }

public static <T extends Comparable<T>> void merge(List<T> list, List<T> left, List<T> right) {

int i = 0, j = 0, k = 0;

while (i < left.size() && j < right.size()) {

list.set(k++, (left.get(i).compareTo(right.get(j)) <= 0) ? left.get(i++) : right.get(j++)); }

while (i < left.size()) list.set(k++, left.get(i++));

while (j < right.size()) list.set(k++, right.get(j++)); }

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the type of list (1 for integers, 2 for strings): ");

int choice = scanner.nextInt();

scanner.nextLine();

if (choice == 1) {

System.out.println("Enter the number of integers:");

A screenshot of a computer

Description automatically generated int n = scanner.nextInt();

List<Integer> integers = new ArrayList<>();

System.out.println("Enter the integers:");

for (int i = 0; i < n; i++) {

integers.add(scanner.nextInt()); }

mergeSort(integers);

System.out.println("Sorted List (Integers): " + integers);

} else if (choice == 2) {

System.out.println("Enter the number of strings:");

int n = scanner.nextInt();

scanner.nextLine();

List<String> strings = new ArrayList<>();

System.out.println("Enter the strings:");

for (int i = 0; i < n; i++) {

strings.add(scanner.nextLine()); }

mergeSort(strings);

System.out.println("Sorted List (Strings): " + strings);

} else {

System.out.println("Invalid choice!");}}

}

**Home Task#04:**

4.You are given a set of bank accounts, each with a unique account number and a balance. Write a Java program to Declare an array of size n to store account balances. Initialize each balance randomly with values between 0 and 100,000. Sort the accounts in descending order of their balances using Quick Sort. Print the sorted list in the format.

**Code: Output:**

import java.util.\*;

A screenshot of a number of accounts

Description automatically generatedpublic class DSALab05 {

static class Account {

int accountNumber;

int balance;

Account(int accountNumber, int balance) {

this.accountNumber = accountNumber;

this.balance = balance; }

public String toString() {

return "Account No. " + accountNumber + " Balance " + balance; } }

public static void quickSort(Account[] accounts, int low, int high) {

if (low < high) {

int pivotIndex = partition(accounts, low, high);

quickSort(accounts, low, pivotIndex - 1);

quickSort(accounts, pivotIndex + 1, high); }

public static int partition(Account[] accounts, int low, int high) {

int pivot = accounts[high].balance;

int i = low - 1;

for (int j = low; j < high; j++) {

if (accounts[j].balance > pivot) {

i++;

swap(accounts, i, j); }

}

swap(accounts, i + 1, high);

return i + 1; }

public static void swap(Account[] accounts, int i, int j) {

Account temp = accounts[i];

accounts[i] = accounts[j];

accounts[j] = temp;}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of accounts: ");

int n = scanner.nextInt();

Account[] accounts = new Account[n];

Random random = new Random();

for (int i = 0; i < n; i++) {

int accountNumber = 1000 + i;

int balance = random.nextInt(100001);

accounts[i] = new Account(accountNumber, balance); }

quickSort(accounts, 0, n - 1);

System.out.println("Sorted Accounts:");

for (Account account : accounts) {

System.out.println(account); }

} }