**22k-5195 LAB 6**

Task 1

import java.util.\*;

public class task1 {

public static int getMax(int a[], int n)

{

int max = a[0];

for (int i = 1; i < n; i++)

if (a[i] > max)

max = a[i];

return max;

}

public static void radix(int a[], int n) {

int m = *getMax*(a, n);

for (int exp = 1; m / exp > 0; exp \*= 10)

*count*(a, n, exp);

}

public static void count(int a[], int n, int exp)

{

int output[] = new int[n];

int i;

int count[] = new int[10];

Arrays.*fill*(count, 0);

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

count[(a[i] / exp) % 10]++;

for (i = 1; i < 10; i++)

count[i] += count[i - 1];

for (i = n - 1; i >= 0; i--) {

output[count[(a[i] / exp) % 10] - 1] = a[i];

count[(a[i] / exp) % 10]--;

}

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

a[i] = output[i];

}

static void display(int a[], int n)

{

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

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

}

public static void main(String[] args)

{

int a[] = { 170, 45, 75, 90, 802, 24, 2, 66 };

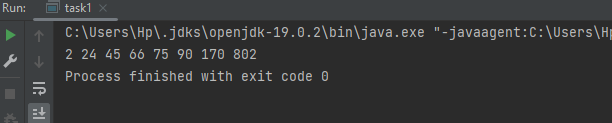
int n = a.length;

*radix*(a, n);

*display*(a, n);

}

}



Task 2

import java.util.Arrays;

public class task2 {

public static int getMax(int a[], int n)

{

int max = a[0];

for (int i = 1; i < n; i++)

if (a[i] > max)

max = a[i];

return max;

}

public static void AscCount(int a[], int exp) {

int n = a.length;

int output[] = new int[n];

int count[] = new int[10];

Arrays.*fill*(count, 0);

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

count[(a[i] / exp) % 10]++;

}

for (int i = 1; i < 10; i++) {

count[i] += count[i - 1];

}

for (int i = n - 1; i >= 0; i--) {

output[count[(a[i] / exp) % 10] - 1] = a[i];

count[(a[i] / exp) % 10]--;

}

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

a[i] = output[i];

}

}

public static void AscRadix(int a[], int n) {

int m = *getMax*(a,n);

for (int exp = 1; m / exp > 0; exp \*= 10) {

*AscCount*(a, exp);

}

}

public static void DescCount(int a[], int exp) {

int n = a.length;

int output[] = new int[n];

int count[] = new int[10];

Arrays.*fill*(count, 0);

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

count[9 - (a[i] / exp) % 10]++;

}

for (int i = 1; i < 10; i++) {

count[i] += count[i - 1];

}

for (int i = n - 1; i >= 0; i--) {

output[count[9 - (a[i] / exp) % 10] - 1] = a[i];

count[9 - (a[i] / exp) % 10]--;

}

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

a[i] = output[i];

}

}

public static void DescRadix(int a[], int n) {

int m = *getMax*(a,n);

for (int exp = 1; m / exp > 0; exp \*= 10) {

*DescCount*(a, exp);

}

}

public static void main(String[] args) {

int a[] = { 170, 45, 75, 90, 802, 24, 2, 66 };

int n = a.length;

System.*out*.println(" Array:");

System.*out*.println(Arrays.*toString*(a));

*AscRadix*(a,n);

System.*out*.println(" Ascending Order:");

System.*out*.println(Arrays.*toString*(a));

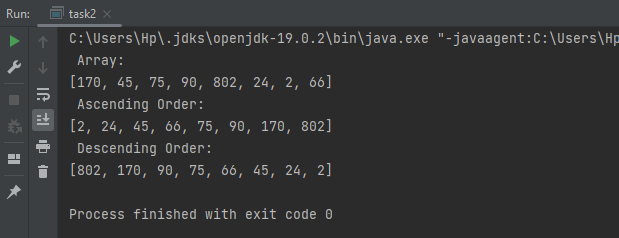
*DescRadix*(a,n);

System.*out*.println(" Descending Order:");

System.*out*.println(Arrays.*toString*(a));

}

}



Task 3

import java.util.Arrays;

public class task3 {

public static int getMax(int a[], int n)

{

int max = a[0];

for (int i = 1; i < n; i++)

if (a[i] > max)

max = a[i];

return max;

}

public static void AscCount(int a[], int exp) {

int n = a.length;

int output[] = new int[n];

int count[] = new int[10];

Arrays.*fill*(count, 0);

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

count[(a[i] / exp) % 10]++;

}

for (int i = 1; i < 10; i++) {

count[i] += count[i - 1];

}

for (int i = n - 1; i >= 0; i--) {

output[count[(a[i] / exp) % 10] - 1] = a[i];

count[(a[i] / exp) % 10]--;

}

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

a[i] = output[i];

}

}

public static void AscRadix(int a[]) {

int n = a.length;

int m = *getMax*(a,n);

for (int exp = 1; m / exp > 0; exp \*= 10) {

*AscCount*(a, exp);

}

}

public static void DescCount(int a[], int exp) {

int n = a.length;

int output[] = new int[n];

int count[] = new int[10];

Arrays.*fill*(count, 0);

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

count[9 - (a[i] / exp) % 10]++;

}

for (int i = 1; i < 10; i++) {

count[i] += count[i - 1];

}

for (int i = n - 1; i >= 0; i--) {

output[count[9 - (a[i] / exp) % 10] - 1] = a[i];

count[9 - (a[i] / exp) % 10]--;

}

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

a[i] = output[i];

}

}

public static void DescRadix(int a[]) {

int n = a.length;

int m = *getMax*(a,n);

for (int exp = 1; m / exp > 0; exp \*= 10) {

*DescCount*(a, exp);

}

}

public static void radix(int a[]) {

int n = a.length;

int odd = 0;

int even = 0;

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

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

even++;

} else {

odd++;

}

}

int[] oddNum = new int[odd];

int[] evenNum = new int[even];

odd = 0;

even = 0;

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

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

evenNum[even++] = a[i];

} else {

oddNum[odd++] = a[i];

}

}

*DescRadix*(oddNum);

*AscRadix*(evenNum);

int i = 0;

int j = 0;

int k = 0;

while (j < odd) {

a[i++] = oddNum[j++];

}

while (k < even) {

a[i++] = evenNum[k++];

}

}

public static void main(String[] args) {

int a[] = { 10, 1, 5, 6, 12, 2, 5, 3, 13, 17, 14 };

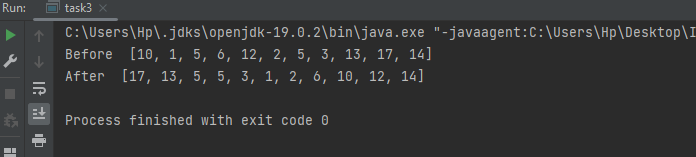
System.*out*.println("Before " + Arrays.*toString*(a));

*radix*(a);

System.*out*.println("After " + Arrays.*toString*(a));

}

}



Task 4

rt java.util.Arrays;

public class task4 {

public static int[] addSort(int[] a, int key) {

int[] newArray = Arrays.*copyOf*(a, a.length + 1);

newArray[newArray.length - 1] = key;

Arrays.*sort*(newArray);

return newArray;

}

public static int Binarysearch(int[] a, int key) {

int left = 0;

int right = a.length - 1;

while (left <= right) {

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

if (a[mid] == key) {

return mid;

} else if (a[mid] < key) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

public static void main(String[] args) {

int[] a = {1, 3, 12, 14, 23, 34, 55, 65, 75, 78};

int rollNum = 5195;

int lastDigits = rollNum % 100;

Arrays.*sort*(a);

System.*out*.println(" Array : " + Arrays.*toString*(a));

int index = *Binarysearch*(a, lastDigits);

if (index == -1) {

a = *addSort*(a, lastDigits);

System.*out*.println("Your value is not found, sorted array: " + Arrays.*toString*(a));

index = *Binarysearch*(a, lastDigits);

}

if (index != -1) {

int answer = a[index];

System.*out*.println("The value matching to the last two digits (95): " + answer);

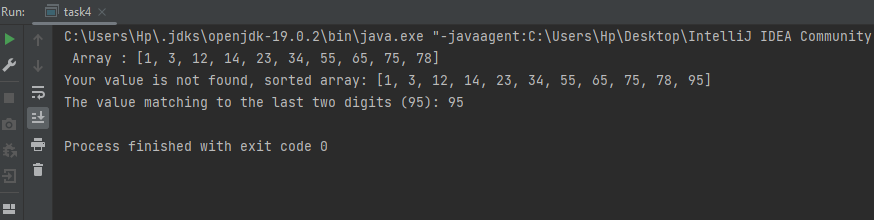
} else {

System.*out*.println("The value matching to the last two digits not found.");

}

}

}



Task 5

import java.util.Scanner;  
  
public class task5 {  
  
 public static int interpolation(int[] a,int hi, int lo, int x){  
 int pos;  
 if (lo<=hi && x>= a[lo] && x<=a[hi] ){  
  
 pos = lo + (((x - a[lo] )\* (hi - lo))/ (a[hi] - a[lo]));  
  
 if(a[pos] == x){  
 return pos;  
 }  
  
 if(a[pos] >x){  
 return *interpolation*(a,hi, pos-1, x);  
 }if(a[pos] <x){  
 return *interpolation*(a,pos+1, lo, x);  
 }  
 }  
  
 return -1;  
 }  
  
  
 public static void main(String[] args) {  
int arr[] = {1,2,3,4,5,6};  
int n =arr.length;  
 Scanner a = new Scanner(System.*in*);  
 System.*out*.println("Enter item to be searched:");  
 int x = a.nextInt();  
 int index = *interpolation*(arr,n-1,0, x);  
  
 if(index != -1){  
 System.*out*.println("Element found at index " + index);  
 }  
 else {  
 System.*out*.println("Element not found");  
 }  
  
  
  
  
  
  
 }  
}

