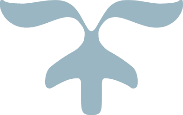


DAA WEEK – 3 SKILL – 3



# [Big Sorting](https://www.hackerrank.com/contests/daa-skill-03-searching-and-sorting-part-1/challenges/big-sorting)

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class Solution {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int n = in.nextInt();

String[] unsorted = new String[n];

for(int i = 0; i < n; i++) unsorted[i] = in.next();

Arrays.sort(unsorted,new Comparator<String>() {

@Override

public int compare(String a, String b)

{

return StringAsIntegerCompare(a,b);

}

});

StringBuilder output = new StringBuilder("");

for(String num : unsorted)

output.append(num+"\n");

System.out.println(output);

}

static int StringAsIntegerCompare(String s1, String s2)

{

if(s1.length() > s2.length()) return 1;

if(s1.length() < s2.length()) return -1;

for(int i = 0; i < s1.length(); i++)

{

if((int)s1.charAt(i) > (int)s2.charAt(i)) return 1;

if((int)s1.charAt(i) < (int)s2.charAt(i)) return -1;

}

return 0;

}

}

**Big Sorting Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Counting Sort 1](https://www.hackerrank.com/contests/daa-skill-03-searching-and-sorting-part-1/challenges/countingsort1)

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

int frequencies[100] = {0};

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

int num;

scanf("%d", &num);

frequencies[num] = frequencies[num] + 1;

}

for (int i = 0; i < 100; i++) {

printf("%d ", frequencies[i]);

}

return 0;

}

**Counting Sort 1 Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Counting Sort 2](https://www.hackerrank.com/contests/daa-skill-03-searching-and-sorting-part-1/challenges/countingsort2)

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

int frequencies[100] = {0};

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

int num;

scanf("%d", &num);

frequencies[num] = frequencies[num] + 1;

}

for (int i = 0; i < 100; i++) {

for (int j = 0; j < frequencies[i]; j++) {

printf("%d ", i);

}

}

return 0;

}

**Counting Sort 2 Test Cases**

**A screenshot of a computer

Description automatically generated**

# [Closest Numbers](https://www.hackerrank.com/contests/daa-skill-03-searching-and-sorting-part-1/challenges/closest-numbers)

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

int compare(const void \*a, const void \*b) {

return (\*(int \*)a - \*(int \*)b);

}

int main() {

int n;

scanf("%d", &n);

int array[n];

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

scanf("%d", &array[i]);

}

qsort(array, n, sizeof(int), compare);

int minAbs = 2147483647;

char pairs[10000] = "";

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

int absDiff;

if ((array[i] < 0 && array[i + 1] < 0) || (array[i] > 0 && array[i + 1] > 0)) {

absDiff = abs(array[i] - array[i + 1]);

} else {

absDiff = abs(array[i]) + abs(array[i + 1]);

}

if (absDiff < minAbs) {

minAbs = absDiff;

sprintf(pairs, "%d %d ", array[i], array[i + 1]);

} else if (absDiff == minAbs) {

char buffer[20];

sprintf(buffer, "%d %d ", array[i], array[i + 1]);

strcat(pairs, buffer);

}

}

printf("%s\n", pairs);

return 0;

}

**Closest Numbers Test Cases**

**A screenshot of a computer

Description automatically generated**

# Find the Median

#include <stdio.h>

#include <stdlib.h>

int compare(const void \*a, const void \*b) {

return (\*(int \*)a - \*(int \*)b);

}

int main() {

int n;

scanf("%d", &n);

int array[n];

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

scanf("%d", &array[i]);

}

qsort(array, n, sizeof(int), compare);

printf("%d\n", array[n / 2]);

return 0;

}

**Find the Median Test Cases**

**A screenshot of a computer

Description automatically generated**

**SKILL WEEK – 3**

[**https://www.hackerrank.com/contests/daa-skill-03-searching-and-sorting-part-1/challenges**](https://www.hackerrank.com/contests/daa-skill-03-searching-and-sorting-part-1/challenges)