**Name : Harshal Chavan College : JSPM’s JSCOE**

**Assignment 1 :**

**package** assignment1;

**import** java.util.Arrays;

**class** Main

{

**private** **static** **void** merge(**int**[] X, **int**[] Y, **int** m, **int** n)

{

**int** k = m + n + 1;

**while** (m >= 0 && n >= 0)

{

**if** (X[m] > Y[n]) {

X[k--] = X[m--];

}

**else** {

X[k--] = Y[n--];

}

}

**while** (n >= 0) {

X[k--] = Y[n--];

}

Arrays.*fill*(Y, 0);

}

**public** **static** **void** rearrange(**int**[] X, **int**[] Y)

{

**if** (X.length == 0) {

**return**;

}

**int** k = 0;

**for** (**int** value: X)

{

**if** (value != 0) {

X[k++] = value;

}

}

*merge*(X, Y, k - 1, Y.length - 1);

}

**public** **static** **void** main (String[] args)

{

**int**[] X = { 0, 2, 0, 3, 0, 5, 6, 0, 0};

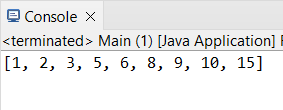
**int**[] Y = { 1, 8, 9, 10, 15 };

*rearrange*(X, Y);

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

}

}

**Output :** 

**Assignment 2 :**

**package** assignment2;

**public** **class** Max\_Sum {

**public** **static** **int** maxSum(**int**[] X, **int**[] Y) {

**int** m = X.length;

**int** n = Y.length;

**int** i = 0, j = 0;

**int** sumX = 0, sumY = 0;

**int** result = 0;

**while** (i < m && j < n) {

**if** (X[i] < Y[j]) {

sumX += X[i++];

} **else** **if** (X[i] > Y[j]) {

sumY += Y[j++];

} **else** {

result += Math.*max*(sumX, sumY);

result += X[i];

sumX = 0;

sumY = 0;

i++;

j++;

}

}

**while** (i < m) {

sumX += X[i++];

}

**while** (j < n) {

sumY += Y[j++];

}

result += Math.*max*(sumX, sumY);

**return** result;

}

**public** **static** **void** main(String[] args) {

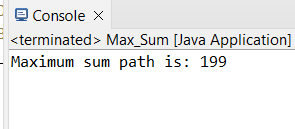
**int**[] X = {3, 6, 7, 8, 10, 12, 15, 18, 100};

**int**[] Y = {1, 2, 3, 5, 7, 9, 10, 11, 15, 16, 18, 25, 50};

System.***out***.println("Maximum sum path is: " + *maxSum*(X, Y));

}

}

**Output :** 

**Assignment 3 :**

**package** assignment3;

**import** java.io.\*;

**import** java.util.HashMap;

**import** java.util.Map;

**public** **class** HashMapEg {

**public** **static** **void** main(String[] args)

{

String str = "Flowers are colourful and Flowers have fragrance";

Map<String, Integer> hashMap = **new** HashMap<>();

String[] words = str.split(" ");

**for** (String word : words) {

Integer integer = hashMap.get(word);

**if** (integer == **null**)

hashMap.put(word, 1);

**else** {

hashMap.put(word, integer + 1);

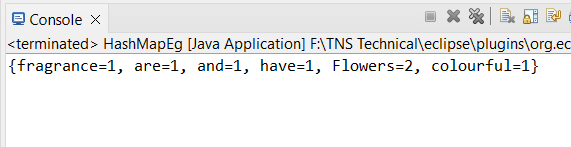
}

}

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

}

}

**Output** : 

**Assignment 4 :**

**package** assignment4;

**import** java.util.HashSet;

**import** java.util.Set;

**public** **class** DupliChar {

**public** **static** **void** main(String[] args) {

String input = "Hello Everybody, Have a nice day";

Set<Character> duplicates = *findDuplicates*(input);

System.***out***.println("Duplicate characters in the string '" + input + "' are:");

**for** (Character ch : duplicates) {

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

}

}

**public** **static** Set<Character> findDuplicates(String str) {

Set<Character> set = **new** HashSet<>();

Set<Character> duplicates = **new** HashSet<>();

**for** (**char** ch : str.toCharArray()) {

**if** (!set.add(ch)) {

duplicates.add(ch);

}

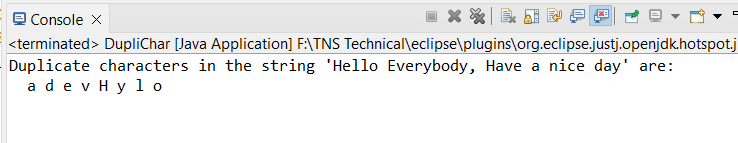
}

**return** duplicates;

}

}

**Output :**

****