1. Java Program to count number of words in a String using HashMap.

import java.io.\*;

import java.util.HashMap;

import java.util.Map;

class GFG {

public static void main(String[] args)

{

// Declaring the String

String str = "Alice is girl and Bob is boy";

// Declaring a HashMap of <String, Integer>

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

// Splitting the words of string

// and storing them in the array.

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

for (String word : words) {

// Asking whether the HashMap contains the

// key or not. Will return null if not.

Integer integer = hashMap.get(word);

if (integer == null)

// Storing the word as key and its

// occurrence as value in the HashMap.

hashMap.put(word, 1);

else {

// Incrementing the value if the word

// is already present in the HashMap.

hashMap.put(word, integer + 1);

}

}

System.out.println(hashMap);

}

}

Output: {Bob=1, Alice=1, and=1, is=2, girl=1, boy=1}

1. 2. **class** PalindromeExample{
2. **public** **static** **void** main(String args[]){
3. **int** r,sum=0,temp;
4. **int** n=454;//It is the number variable to be checked for palindrome
6. temp=n;
7. **while**(n>0){
8. r=n%10;  //getting remainder
9. sum=(sum\*10)+r;
10. n=n/10;
11. }
12. **if**(temp==sum)
13. System.out.println("palindrome number ");
14. **else**
15. System.out.println("not palindrome");
16. }
17. }

Output:

palindrome number

3.

1. **class** FibonacciExample1{
2. **public** **static** **void** main(String args[])
3. {
4. **int** n1=0,n2=1,n3,i,count=10;
5. System.out.print(n1+" "+n2);//printing 0 and 1
7. **for**(i=2;i<count;++i)//loop starts from 2 because 0 and 1 are already printed
8. {
9. n3=n1+n2;
10. System.out.print(" "+n3);
11. n1=n2;
12. n2=n3;
13. }
15. }}

Output: 0 1 1 2 3 5 8 13 21 34

4. Duplicate charcters in a string

1. **public** **class** DuplicateCharacters {
2. **public** **static** **void** main(String[] args) {
3. String string1 = "Great responsibility";
4. **int** count;
6. //Converts given string into character array
7. **char** string[] = string1.toCharArray();
9. System.out.println("Duplicate characters in a given string: ");
10. //Counts each character present in the string
11. **for**(**int** i = 0; i <string.length; i++) {
12. count = 1;
13. **for**(**int** j = i+1; j <string.length; j++) {
14. **if**(string[i] == string[j] && string[i] != ' ') {
15. count++;
16. //Set string[j] to 0 to avoid printing visited character
17. string[j] = '0';
18. }
19. }
20. //A character is considered as duplicate if count is greater than 1
21. **if**(count > 1 && string[i] != '0')
22. System.out.println(string[i]);
23. }
24. }
25. }

**Output:**

Duplicate characters in a given string:

r

e

t

s

i

5.Second Highest number in array

public class ThirdLargestNumberInAnArray {

   public static void main(String args[]){

      int temp, size;

      int array[] = {10, 20, 25, 63, 96, 57};

      size = array.length;

      for(int i = 0; i<size; i++ ){

         for(int j = i+1; j<size; j++){

            if(array[i]>array[j]){

               temp = array[i];

               array[i] = array[j];

               array[j] = temp;

            }

         }

      }

      System.*out*.println("Third second largest number is:: "+array[size-2]);

   }

}

Output

Third second largest number is:: 63