**LAB 1**

1. Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Robert Brett Roser, then the output should be R.B.Roser.

**package** p1;

**import** java.util.\*;

**import** java.util.Scanner;

**public** **class** q\_1 {

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

String firstName, middleName,lastName;

System.***out***.print("Enter First Name: ");

Scanner sc = **new** Scanner(System.***in***);

firstName = sc.next();

System.***out***.print("Enter Middle Name: ");

Scanner sc1 = **new** Scanner(System.***in***);

middleName = sc1.next();

System.***out***.print("Enter Last Name: ");

Scanner sc2 = **new** Scanner(System.***in***);

lastName = sc2.next();

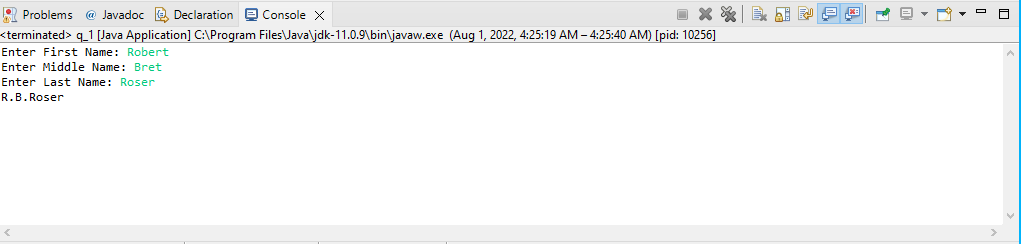
**char** f1 = firstName.charAt(0);

**char** m1 = middleName.charAt(0);

System.***out***.print(f1+"."+m1+"."+lastName);

}

}



1. Input a string of alphabets. Find out the number of occurrence of all alphabets in that string. Find out the alphabet with maximum occurrence.

**package** proj1;

**public** **class** operation

{

**static** **final** **int** ***ASCII\_SIZE*** = 256;

**static** **char** getMaxOccurringChar(String str)

{

**int** count[] = **new** **int**[***ASCII\_SIZE***];

**int** len = str.length();

**for** (**int** i=0; i<len; i++)

count[str.charAt(i)]++;

**int** max = -1;

**char** result = ' ';

**for** (**int** i = 0; i < len; i++) {

**if** (max < count[str.charAt(i)]) {

max = count[str.charAt(i)];

result = str.charAt(i);

}

}

**return** result;

}

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

{

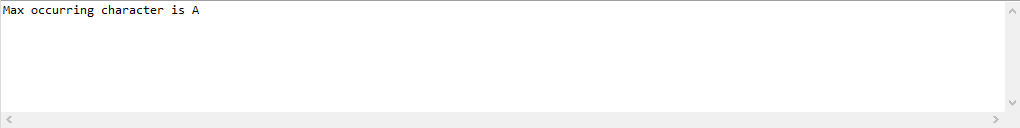
String str = "Ahmer Aziz";

System.***out***.println("Max occurring character is " +

*getMaxOccurringChar*(str));

}

}



1. Write a program to find out longest palindrome in a given string?

**Input:** Given string: “forgeeksskeegfor",

**Output:** "geeksskeeg".

**package** p1;

**import** java.util.\*;

**import** java.util.Scanner;

**public** **class** q\_1 {

**static** **void** printSubStr(String str, **int** low, **int** high)

{

**for** (**int** i = low; i <= high; ++i)

System.***out***.print(str.charAt(i));

}

**static** **int** longestPalSubstr(String str)

{

**int** n = str.length();

**int** maxLength = 1, start = 0;

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

{

**for** (**int** j = i; j < str.length(); j++)

{

**int** flag = 1;

**for** (**int** k = 0; k < (j - i + 1) / 2; k++)

**if** (str.charAt(i + k) != str.charAt(j - k))

flag = 0;

**if** (flag!=0 && (j - i + 1) > maxLength)

{

start = i;

maxLength = j - i + 1;

}

}

}

System.***out***.print("Longest palindrome subString is: ");

*printSubStr*(str, start, start + maxLength - 1);

**return** maxLength;

}

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

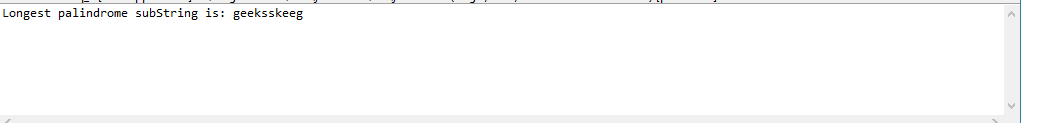
{

String str = "forgeeksskeegfor";

*longestPalSubstr*(str);

}

}



1. Write a program to find first non-repeating character from a string.

**Input:** Given string:" geeksforgeeks",

**Output:** "f".

**package** p1;

**import** java.util.\*;

**import** java.util.Scanner;

**public** **class** q\_1 {

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

String string1 = "geeksforgeeks";

String str = **new** String();

**int** count;

**char** string[] = string1.toCharArray();

System.***out***.println("First non-repeating character from a string: ");

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

{

count = 1;

**for**(**int** j = i+1; j <string.length; j++)

{

**if**(string[i] == string[j] && string[i] != ' ')

{

count++;

string[j] = '0';

}

}

**if**(count == 1 && string[i] != '0')

{

str = str + string[i];

}

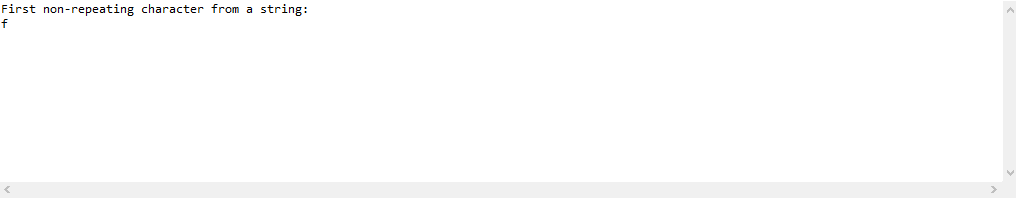
}

**char** ch=str.charAt(0);

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

}

}



1. In an array 1-100 numbers are stored, one number is missing write a program to find it.

**package** proj1;

**import** java.util.\*;

**import** java.util.Scanner;

**public** **class** q\_1 {

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

{

**int** i,n,sum = 0,missing=0;

System.***out***.print("Enter size of the array: ");

Scanner sc = **new** Scanner(System.***in***);

n=sc.nextInt();

**int**[]arr=**new** **int**[n-1];

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

{

arr[i]=sc.nextInt();

sum=sum+arr[i];

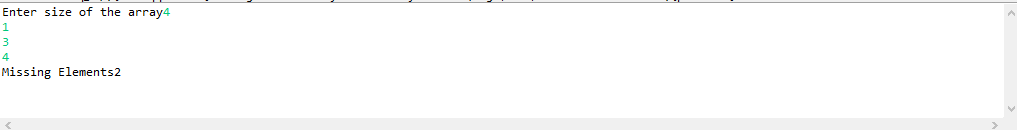
}

missing=(n\*(n+1))/2-sum;

System.***out***.print("Missing Elements: "+missing);

}

}



1. Write a program to remove duplicate characters from String.  
   **package** p1;

**import** java.util.\*;

**import** java.util.Scanner;

**public** **class** q\_1 {

**static** String removeDuplicate(**char** str[], **int** n)

{

**int** index = 0;

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

{

**int** j;

**for** (j = 0; j < i; j++)

{

**if** (str[i] == str[j])

{

**break**;

}

}

**if** (j == i)

{

str[index++] = str[i];

}

}

**return** String.*valueOf*(Arrays.*copyOf*(str, index));

}

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

{

**char** str[] = "Ahmer Aziz".toCharArray();

**int** n = str.length;

System.***out***.println(*removeDuplicate*(str, n));

}

}

