11 Largest Key in HashMap

Write a program that construts a hashmap and returns the value corresponding to the largest key.

Include a class UserMainCode with a static method **getMaxKeyValue** which accepts a string. The return type (String) should be the value corresponding to the largest key.

Create a Class Main which would be used to accept Input string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of 2n+1 values. The first value corresponds to size of the hashmap. The next n pair of numbers equals the integer key and value as string.

Output consists of a string which is the value of largest key.

Refer sample output for formatting specifications.

Sample Input 1:

3

12

amron

9

Exide

7

SF

Sample Output 1:

amron

Solutions:

```
import java.util.HashMap;
import java.util.Scanner;

public class Main {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
```

```
int n= sc.nextInt();
HashMap<Integer,String> hm= new HashMap<Integer,String>();
for(int i=0;i<n;i++)
        hm.put(sc.nextInt(), sc.next());
System.out.println(User.getMaxKeyValue(hm));

}
}
```

```
importjava.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstaticString getMaxKeyValue(HashMap<Integer,String> hm){
      int max=0;
      String nn=null;
      Iterator<Integer> it = hm.keySet().iterator();
      while(it.hasNext())
      {
            int key=it.next();
            String name=hm.get(key);
            if(key>max)
            {
                  key=max;
            nn=name;
            }
      }
      return nn;
      }
}
```

12. All Numbers

Write a program to read a string array and return 1 if all the elements of the array are numbers, else return -1.

Include a class UserMainCode with a static method **validateNumber** which accepts a string aray. The return type (integer) should be -1 or 1 based on the above rules.

Create a Class Main which would be used to accept Input string array and call the static method present in UserMainCode.
The string array is said to be valid if all the elements in the array are numbers. Else it is invalid.

Input and Output Format:

Input consists of an integer specifying the size of string array followed by n strings.

Refer sample output for formatting specifications.

Sample Input 1:

4
123

24.5

23

one

Sample Output 1:

invalid

Sample Input 2:

2
123

24.5

Sample Output 2:

valid

```
import java.util.HashMap;

import java.util.Scanner;



public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n= sc.nextInt();

String[] s= new String[n];

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

```
        s[i]=sc.next();

int res=User.validateNumber(s);

if(res==1)

        System.out.println("Valid");

else

        System.out.println("invalid");



}

}
```

```
importjava.util.ArrayList;
importjava.util.HashMap;
importjava.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstaticint validateNumber(String s[]){
      int res=0;
      int count=0,temp=0;
      String s1=null;
      for(int i=0;i<s.length;i++)
      {
            s1=s[i];
            count=0;
            for(int j=0;j<s1.length();j++)
            {

            if(s1.charAt(j)>='0'&& s1.charAt(j)<='9' || s1.charAt(j)=='.')
                  count++;
            }
      if(count==s1.length())
            temp++;
      }
      if(temp==s.length)
            res=1;
      else
            res=-1;
      return res;
      }
}
```

13. Day of the Week

Write a program to read a date as string (MM-dd-yyyy) and return the day of week on that date.

Include a class UserMainCode with a static method **getDay** which accepts the string. The return type (string) should be the day of the week.

Create a Class Main which would be used to accept Input string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a string.

Output consists of a string.

Refer sample output for formatting specifications.

Sample Input 1:

07-13-2012

Sample Output 1:

Friday

Solutions :

User :

```
import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;


public class User

{

public static String calculateBornDay(String d) throws ParseException

{
```

```
SimpleDateFormat sdf= new SimpleDateFormat("MM-dd-yyyy");

SimpleDateFormat s= new SimpleDateFormat("EEEE");

Date d1= new Date();

d1= sdf.parse(d);

String day=s.format(d1);

return day;

}

}
```

14. Max Substring

Write a program to accept two string inputs. The first being a source string and second one a delimiter. The source string contains the delimiter at various locations. Your job is to return the substring with maximum number of characters. If two or more substrings have maximim number of characters return the substring which appears first. The size of the delimiter is 1.

Include a class UserMainCode with a static method **extractMax** which accepts the string. The return type (string) should be the max substring.

Create a Class Main which would be used to accept Input string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a source string and delimiter.

Output consists of a string.

Refer sample output for formatting specifications.

Sample Input 1:

delhi-pune-patna

-

Sample Output 1:

Delhi

```
importjava.util.HashMap;
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = new Scanner(System.in);
String s1=sc.next();
String s2=sc.next();
System.out.println(User.extractMax(s1,s2));

}
}
```

```
importjava.util.ArrayList;
importjava.util.HashMap;
importjava.util.Iterator;
import java.util.StringTokenizer;


publicclass User {
publicstatic String extractMax(String s1,String s2){
	StringTokenizer st= new StringTokenizer(s1,s2);
	int max=0,c=0;
	String str=null;
	while(st.hasMoreTokens())
	{
		String s= st.nextToken();
		c=s.length();
		if(c>max)
		{
			c=max;
		str=s;
		}

	}
	return str;

	}
}
```

15. States and Capitals

Write a program that construts a hashmap with “state” as key and “capital” as its value. If the next input is a state, then it should return capital$state in lowercase.

Include a class UserMainCode with a static method **getCapital** which accepts a hashmap. The return type is the string as given in the above statement

Create a Class Main which would be used to accept Input string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of 2n+2 values. The first value corresponds to size of the hashmap. The next n pair of numbers contains the state and capital. The last value consists of the “state” input.

Output consists of a string as mentioned in the problem statement.

Refer sample output for formatting specifications.

Sample Input 1:

3

Karnataka

Bangaluru

Punjab

Chandigarh

Gujarat

Gandhinagar

Punjab

Sample Output 1:

chandigarh$punjab

```
 import java.util.HashMap;
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = new Scanner(System.in);
int n=sc.nextInt();
HashMap<String,String> hm= new HashMap<String,String>();
for(int i=0;i<n;i++)
      hm.put(sc.next(),sc.next());
String s=sc.next();
System.out.println(User.getCapital(hm,s));

}
}
```

```
importjava.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstatic String getCapital(HashMap<String,String> hm,String s){
	Iterator<String> it=hm.keySet().iterator();
	StringBuffer sb= new StringBuffer();
	while(it.hasNext())
	{
		String state=it.next();
		String cap=hm.get(state);
		if(state.equalsIgnoreCase(s))
		{
			sb.append(cap).append('$').append(state);
		}
	}
	return sb.toString().toLowerCase();

	}
}
```

16. Simple String Manipulation - II

Write a program to read a string and return an integer based on the following rules.

If the first word and the last word in the String match, then return the number of characters in the word else return sum of the characters in both words. Assume the Strings to be case - sensitive.

Include a class UserMainCode with a static method **calculateWordSum** which accepts a string. The return type (integer) should be based on the above rules.

Create a Class Main which would be used to accept Input string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a string with maximum size of 100 characters.

Output consists of a string.

Refer sample output for formatting specifications.

Sample Input 1:

COGNIZANT TECHNOLOGY SOLUTIONS COGNIZANT

Sample Output 1:

9

Sample Input 2:

HOW ARE YOU

Sample Output 2:

6

```
importjava.util.HashMap;
import java.util.Scanner;

publicclass Main {
publicstaticvoid main(String[] args) {
Scanner sc = new Scanner(System.in);

String s=sc.nextLine();
System.out.println(User.calculateWordSum (s));

}
}
```

```
importjava.util.ArrayList;
importjava.util.HashMap;
importjava.util.Iterator;
import java.util.StringTokenizer;


publicclass User {
publicstaticint calculateWordSum (String s){
	int sum=0,i=0;
	StringTokenizer st=new StringTokenizer(s," ");
	String[] s1= new String[st.countTokens()];
	while(st.hasMoreTokens())
	{
	s1[i]=st.nextToken();
	i++;
	}
	if(s1[0].equals(s1[s1.length-1]))
		sum=s1[0].length();
	else
		sum=s1[0].length()+s1[s1.length-1].length();
	return sum;
	}
}
```

17. Vowels, Arrays & ArrayLists

Write a program to read an array of strings and return an arraylist which consists of words whose both first and last characters are vowels. Assume all inputs are in lowecase.

Include a class UserMainCode with a static method **matchCharacter** which accepts a string array. The return type shoud be an arraylist which should contain elements as mentioned above.

Create a Class Main which would be used to accept Input array and call the static method present in UserMainCode.

Input and Output Format:

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' string correspond to the elements in the array.

Output consists of strings which are elements of arraylist

Refer sample output for formatting specifications.

Sample Input 1:

4

abcde

pqrs

abci

orto

Sample Output 1:

abcde

abci

orto

```
import java.util.HashMap;
import java.util.Scanner;
 public class Main {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
```

```
int n=sc.nextInt();
String[] s= new String[n];
for(int i=0;i<n;i++)
      s[i]=sc.next();
System.out.println(User.matchCharacter  (s));
}
}

import java.util.ArrayList;
importjava.util.HashMap;
importjava.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstatic ArrayList<String> matchCharacter  (String[] s){
      ArrayList<String> a= new ArrayList<String>();
      for(int i=0;i<s.length;i++)
      {
            System.out.println(s[i].charAt(0));
            System.out.println(s[i].charAt(s[i].length()-1));
            if((s[i].charAt(0)=='a'|| s[i].charAt(0)=='e'||
                        s[i].charAt(0)=='i'||s[i].charAt(0)=='o'||
                        s[i].charAt(0)=='u') && (s[i].charAt(s[i].length()-
1)=='a'||
                                          s[i].charAt(s[i].length()-
1)=='e'||s[i].charAt(s[i].length()-1)=='i'||
                                          s[i].charAt(s[i].length()-
1)=='o'||s[i].charAt(s[i].length()-1)=='u'))
            {
                  a.add(s[i]);
            }
      }
      return a;
      }
}
```

18.Transfer from Hashmap to Arraylist

Write a program that constructs a hashmap with “employee id” as key and “name” as its value. Based on the rules below, on being satisfied, the name must be added to the arraylist.

i)First character should be small and the last character should be Capital.

ii)In name at least one digit should be there.

Include a class UserMainCode with a static method **getName** which accepts a hashmap. The return type is an arraylist as expected in the above statement

Create a Class Main which would be used to accept Input string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of 2n+1 values. The first value corresponds to size of the hashmap. The next n pair of numbers contains the employee id and name.

Output consists of arraylist of strings as mentioned in the problem statement.

Refer sample output for formatting specifications.

Sample Input 1:

4

1

ravi5raJ

2

sita8gitA

3

ram8sitA

4

rahul

Sample Output 1:

ravi5raJ

sita8gitA

ram8sitA

```
import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;



public class Main {
```

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

Scanner sc = new Scanner(System.in);

int n=sc.nextInt();

HashMap<Integer,String> hm= new HashMap<Integer,String>();

ArrayList<String> a= new ArrayList<String>();

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

        hm.put(sc.nextInt(), sc.next());

a=User.getName(hm);

for(int i=0;i<a.size();i++)

{

        System.out.println(a.get(i));

}

}

}
```

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstatic ArrayList<String> getName(HashMap<Integer,String> hm){
      ArrayList<String> a= new ArrayList<String>();
      Iterator<Integer> it=hm.keySet().iterator();
      while(it.hasNext())
      {
            int id=it.next();
            String name=hm.get(id);
            for(int i=0;i<name.length();i++)
            {
            if(name.charAt(0)>=97 && name.charAt(0)<=122 &&
```

```
                        name.charAt(name.length()-1)>=65 &&
name.charAt(name.length()-1)<=96)
            {
                  if(name.charAt(i)>='0'&& name.charAt(i)<='9')
                  {
                        a.add(name);
                  }
            }

            }
      }
      return a;
      }
}
```

19. Max Admissions

Write a program that reads details about number of admissions per year of a particular college, return the year which had maximum admissions. The details are stored in an arraylist with the first index being year and next being admissions count.

Include a class UserMainCode with a static method **getYear** which accepts a arraylist. The return type is an integer indicating the year of max admissions.

Create a Class Main which would be used to accept Input string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of 2n+1 values. The first value corresponds to size of the data (year & admissions). The next n pair of numbers contains the year and admissions count.

Output consists of an integer as mentioned in the problem statement.

Refer sample output for formatting specifications.

Sample Input 1:

4

2010

200000

2011

300000

2012

45000

2013

25000

Sample Output 1:

2011

```
import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;



public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n=sc.nextInt();

ArrayList<Integer> a= new ArrayList<Integer>();

for(int i=0;i<n*2;i++)

        a.add(sc.nextInt());



        System.out.println(User.getYear(a));



}

}
```

```
import java.util.ArrayList;
importjava.util.HashMap;
```

```
importjava.util.Iterator;
importjava.util.StringTokenizer;

publicclass User {
publicstaticint getYear(ArrayList<Integer> a){
	intyear=0;
	int max=0;
	for(int i=1;i<a.size();i=i+2)
	{
		int x=a.get(i);
		if(x>max)
		{
			max=x;
			year=a.get(i-1);
		}
	}
	returnyear;
	}
}
```

20. Sum Non Prime Numbers

Write a program to calculate the sum of all the non prime positive numbers less than or equal to the given number.

Note: prime is a natural number greater than 1 that has no positive divisors other than 1 and itself

Example:
input = 9
Prime numbers = 2,3,5 and 7
output = 1+4+6+8+9=28

Include a class **UserMainCode** with a static method "**addNumbers**" that accepts an integer arguement and returns an integer.

Create a class **Main** which would get an integer as input and call the static method **validateNumber** present in the UserMainCode.

Input and Output Format:
Input consists of an integer.
Output consists of an integer.

Sample Input:
9

Sample Output:

28

```
import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;



public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n=sc.nextInt();

        System.out.println(User.addNumbers(n));



}

}
```

```
importjava.util.ArrayList;
importjava.util.HashMap;
importjava.util.Iterator;
importjava.util.StringTokenizer;


publicclass User {
publicstaticint addNumbers(int n){
      int c=0,sum=0;
      for (int i = 1; i <= n; i++)
     {
        c=0;
for(int j=1;j<=i;j++)
        {
if(i%j==0)
            c++;
        }
```

```
        if (c==2)
                ;
        else
          sum=sum+i;
       }
        return sum;
        }
}
```
