**ST-2 Core Java (CSL5303) SET B**

**MCQs**

1. Determine the output of following Java Code.

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

**publicclass**mcq{

**publicstaticvoid** main(String args[]) {

TreeSet<String> tree = **new**TreeSet<String>();

tree.add("3");

tree.add("9");

tree.add("1");

tree.add("4");

tree.add("8");

System.*out*.println(tree);

}

}

1. **[1, 3, 4, 8, 9]**
2. {1, 3, 4, 8, 9}
3. {3, 9, 1, 4, 8}
4. [3, 9, 1, 4, 8]
5. How many threads are created in following code?

public class ThreadExtended extends Thread {

public void run() {

System.out.println("\nThread is running now\n");

}

public static void main(String[] args) {

ThreadExtendedthreadE = new ThreadExtended();

threadE.start();

}

}

1. 0
2. 1
3. **2**
4. 3
5. Examine the following Java Code

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

**publicclass**mcq{

**publicstaticvoid** main(String args[]) {

ArrayList<String> list = **new**ArrayList<String> ();

list.add( "Andy" );

list.add( "Bart" );

list.add( "Carl" );

list.add( "Doug" );

list.add( "Elmo" );

System.*out*.println(list);

}

}

Which of the following will change the list so that it looks like:

**[Andy, Bart, Carl, Doug]**

1. list.remove( list.size() );
2. **list.remove( list.size()-1);**
3. list.remove( 5 );
4. list.clear( "Elmo" );
5. What is the output of following code if input is given as: “xyzsquad”?

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

**publicclass**mcq{

**publicstaticvoid** main(String[] args) **throws** Exception{

**char** c;

BufferedReaderobj = **new**BufferedReader(**new**InputStreamReader(System.*in*));

**do**{

c = (**char**) obj.read();

System.*out*.print(c);

} **while**(c != 'q');

}

}

1. xyzs
2. **xyzsq**
3. xyzsqu
4. None of the above
5. What happens if we put a key object in a HashMap which exists?
6. **The new object replaces the older object**
7. The new object is discarded
8. The old object is removed from the map
9. It throws an exception as the key already exists in the map
10. Which of these types cannot be used to initiate a generic type?
11. Integer class
12. Float class
13. **Primitive Types**
14. Collections
15. Determine the output of following Java Code.

importjava.util.\*;

public class mcq{

public static void main(String args[]) {

Vector<Integer>obj = new Vector<Integer>(5,2);

obj.addElement(3);

obj.addElement(1);

obj.addElement(5);

System.out.println(obj.capacity());

}

}

1. 10
2. **5**
3. 3
4. None of the above
5. A priority queue can efficiently be implemented using which of the following data structures?
6. Array
7. LinkedList
8. **Heap**
9. None of the above.
10. What will be the output of following code?

**publicclass**mcq**extends** Thread{

**publicstaticvoid** main(String [] args) {

mcq t = **new**mcq();

t.start();

}

**publicvoid** run() {

**for**(**int** i = 0; i < 3; ++i) {

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

}

}

}

1. 0..1..2..3..
2. 1..2..3..
3. Compilation fails
4. **0..1..2..**
5. What is the relation between hashset and hashmap?
6. **HashSet internally implements HashMap**
7. HashMap internally implements HashSet
8. HashMap is the interface; HashSet is the concrete class
9. HashSet is the interface; HashMap is the concrete class
10. What will be the output of following code?

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

**publicclass**mcq {

**publicstaticvoid** main(String args[]) {

List<Integer> l = **new**LinkedList<Integer>();

l.add(2);

l.add(3);

l.add(4);

List<Integer> s=**new**LinkedList<Integer>();

s.add(7);

s.add(8);

s.add(9);

**for** (Iterator<Integer> itr1=l.iterator(); itr1.hasNext();){

**for** (Iterator<Integer> itr2=s.iterator(); itr2.hasNext();) {

**if** (itr1.next() < itr2.next())

System.*out*.print(itr1.next());

}

}

}

}

1. 2 3 4 7 8 9
2. **Exception**
3. 2 2 2 3 3 3 4 4 4
4. 2 3 4
5. Predict the output of following java code.

import java.io.\*;

public class mcq{

public static void main(String[] args){

String obj = "sessional";

int length = obj.length();

char c[] = new char[length];

obj.getChars(0, length, c, 0);

CharArrayReader input1 = new CharArrayReader(c);

CharArrayReader input2 = new CharArrayReader(c, 0, 4);

inti,j;

try {

while ((i = input1.read()) == (j = input2.read())) {

System.out.print((char)i);

}

}

catch (IOException e) {

e.printStackTrace();

}

}

}

1. s
2. se
3. **sess**
4. ses
5. What will be the output of following code?

class Line {

synchronized public void getLine() {

for (int i = 0; i < 3; i++) {

System.out.print(i+” “);

try {

Thread.sleep(400);

}

catch (Exception e) {

System.out.println(e);

}

}

}

}

class Train extends Thread {

Line line;

Train(Line line){

this.line = line;

}

public void run() {

line.getLine();

}

}

public class mcq {

public static void main(String[] args) {

Line obj = new Line();

Train train1 = new Train(obj);

Train train2 = new Train(obj);

train1.start();

train2.start();

}

}

1. 0 0 1 1 2 2
2. 0 1 0 1 2 2
3. 0 1 1 0 2 2
4. **0 1 2 0 1 2**
5. What will be the output of following code?

importjava.util.\*;

classgenericstack<E>{

Stack <E>stk = new Stack <E>();

public void push(E obj) {

stk.push(obj);

}

public E pop() {

E obj = stk.pop();

returnobj;

}

}

public class mcq{

public static void main(String args[]) {

genericstack<String>gs = new genericstack<String>();

gs.push("Java");

System.out.print(gs.pop() + " ");

genericstack<Integer> gs1 = new genericstack<Integer>();

gs1.push(1234);

System.out.println(gs1.pop());

}

}

1. 1234 Java
2. Java
3. **Java 1234**
4. 1234
5. What is the output of below snippet?

importjava.util.\*;

public class mcq{

public static void main(String[] args) {

Map<Integer, Object>sampleMap = new TreeMap<Integer, Object>();

sampleMap.put(1, null);

sampleMap.put(5, null);

sampleMap.put(3, null);

sampleMap.put(2, null);

sampleMap.put(4, null);

System.out.println(sampleMap);

}

}

1. **{1=null, 2=null, 3=null, 4=null, 5=null**}
2. {5=null}
3. Exception is thrown
4. {1=null, 5=null, 3=null, 2=null, 4=null}
5. ***Print Bracket Number***

Given a string of length n consisting of round brackets. The task is to print the bracket numbers when the string is being parsed.

Input Format

The first line contains an integer T, the number of test cases. For each test case, there is a string containing round brackets.

*(Refer Sample Input below for the explanation)*

Output Format

For each test case, the output is the bracket numbers of the string.

*(Refer Sample Output below for the explanation)*

Sample Input1:

3​

(())()​

((())(()))

((((()

Sample Output1:

1 2 2 1 3 3

1 2 3 3 2 4 5 5 4 1

1 2 3 4 5 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2  (((((  ()()( | 1  ((()()()(((()))) | 3  ()()()()()()()()()()  (((((())))))((((  (((())(( | 1  ((((((((((((((()))))))))) | 4  (((())  ()(())(())  ((()))((()))  ((((((())))((())) |
| 1 2 3 4 5  1 1 2 2 3 | 1 2 3 3 4 4 5 5 6 7 8 9 9 8 7 6 | 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10  1 2 3 4 5 6 6 5 4 3 2 1 7 8 9 10  1 2 3 4 4 3 5 6 | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 15 14 13 12 11 10 9 8 7 6 | 1 2 3 4 4 3  1 1 2 3 3 2 4 5 5 4  1 2 3 3 2 1 4 5 6 6 5 4  1 2 3 4 5 6 7 7 6 5 4 8 9 10 10 9 8 |

*Solution:*

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

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

**publicclass** file

{

**staticvoid**printBracketNumber(String exp,**int** n)

{

**intleft\_bnum = 1;**

**Stack<Integer>right\_bnum =new Stack<Integer>();**

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

**{**

**if (exp.charAt(i) == '(')**

**{**

**System.*out*.print(left\_bnum + " ");**

**right\_bnum.push(left\_bnum);**

**left\_bnum++;**

**}**

**elseif(exp.charAt(i) == ')')**

**{**

**System.*out*.print(right\_bnum.peek() + " ");**

**right\_bnum.pop();**

**}**

**}**

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

}

**publicstaticvoid** main(String args[])

{

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

**int** t=scan.nextInt();

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

String exp = scan.next();

*printBracketNumber*(exp, exp.length());

}

}

}

1. ***Marks/Name Sort***

It is important for a teacher to continuous keep analysing students’ performance and give them feedback so students can improve. After sessional test II, teacher has got student report which contains student name and marks in Java course. For analysing teacher required sorted list either on base of name or marks.

Help them find the students’ performance by sorting and display the student list as per requirement. If “MarksSort” is enter then list should be sorted with marks in the descending order and if “NameSort” is entered as choice, then list should be sorted by name of students in alphabetic order. If any two or more students have same name, then list further sorted by marks in descending order for particularly those students only. Similarly, if any two or more students have same marks then list further sorted by name in alphabetic order. Write a program to complete analysis process.

Input Format

The first line of input contains numbers of students, n.

The next lines contain name and marks of every student.

The last line of input contains a choice whether to sort as MarksSort or NameSort.

*(Refer Sample Input below for the explanation)*

Output Format

Output contains serial no, name and marks according to sorting choice entered.

*(Refer Sample Output below for the explanation)*

Sample Input 1

6

Haritha

25

Shanmathi

23

Thara

26

Gopal

65

Krishna

45

Lakshmi

35

MarksSort

Sample Output 1

1 Gopal 65

2 Krishna 45

3 Lakshmi 35

4 Thara 26

5 Haritha 25

6 Shanmathi 23

Sample Input 2

6

Haritha

25

Shanmathi

23

Thara

26

Shanmathi

65

Krishna

45

Lakshmi

35

NameSort

Sample Output 2

1 Haritha 25

2 Krishna 45

3 Lakshmi 35

4 Shanmathi 65

5 Shanmathi 23

6 Thara 26

Sample Input 3

4

Haritha

25

Shanmathi

23

Thara

26

Gopal

23

MarksSort

Sample Output 3

1 Thara 26

2 Haritha 25

3 Gopal 23

4 Shanmathi 23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case1 | Test case2 | Test case3 | Test case4 | Test Case5 |
| 7 Haritha 25 Shanmathi 23 Thara 26 Gopal 65 Krishna 45 Lakshmi 35  Anu  35  MarksSort | 5 Haritha 25 Shanmathi 23 Thara 26 Gopal 23 Krishna Rana 45 MarksSort | 5 Haritha 25 Shanmathi 73 Thara 26 Shanmathi 65 Krishna Rana 45 NameSort | 5 Haritha 25 Shanmathi 23 Thara 26 Krishna 65 Krishna 45  NameSort | 7 Haritha 25 Shanmathi 23 Thara 26 GopalPandit 65 Krishna 45 Lakshmi 35  GopalPandit  77  NameSort |
| 1 Gopal 65  2 Krishna 45  3 Anu 35  4 Lakshmi 35  5 Thara 26  6 Haritha 25  7 Shanmathi 23 | 1 Krishna Rana 45  2 Thara 26  3 Haritha 25  4 Gopal 23  5 Shanmathi 23 | 1 Haritha 25  2 Krishna Rana 45  3 Shanmathi 73  4 Shanmathi 65  5 Thara 26 | 1 Haritha 25  2 Krishna 65  3 Krishna 45  4 Shanmathi 23  5 Thara 26 | 1 GopalPandit 77  2 GopalPandit 65  3 Haritha 25  4 Krishna 45  5 Lakshmi 35  6 Shanmathi 23  7 Thara 26 |

*Solution*:

import java.io.\*;

importjava.util.\*;

class Passenger{

**int age;**

**String name;**

**Passenger(String name,int age){**

**this.age=age;**

**this.name=name;**

**}**

**public String toString(){**

**return this.name+" "+this.age;**

**}**

**}**

**classSortByAge implements Comparator<Passenger>{**

**publicint compare(Passenger p1,Passenger p2){**

**if(p1.age==p2.age)**

**return (p1.name.compareTo(p2.name));**

**else**

**return -(p1.age-p2.age);**

**}**

**}**

**classSortByName implements Comparator<Passenger>{**

**publicint compare(Passenger p1,Passenger p2) {**

**if((p1.name).equals(p2.name))**

**return -(p1.age-p2.age);**

**else**

**return ( p1.name.compareTo(p2.name));**

**}**

**}**

public class file{

public static void main(String[] args) throws IOException{

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

List<Passenger> l1=new LinkedList<Passenger>();

int size1=Integer.parseInt(reader.readLine());

for(int i=0;i<size1;i++) {

String name=reader.readLine();

int age=Integer.parseInt(reader.readLine());

l1.add(new Passenger(name, age));

}

String choice=reader.readLine();

if(choice.equals("MarksSort"))

Collections.sort(l1,new SortByAge());

else if(choice.equals("NameSort")) {

Collections.sort(l1,new SortByName());

}

Iterator <Passenger>it=l1.iterator();

int i=1;

while(it.hasNext())

{

Passenger obj=(Passenger)it.next();

System.out.println(i++ +" "+obj.name+" "+obj.age);

}

}

}

***18. Character Frequency***

Write a Java program to calculate the character frequency in an announcement made in the airport. The input consists of a single sentence and the output displays a graphical chart displaying the frequency of each character by number of asterisk (\*). Display the characters in the output in reverse alphabetical order. Compute the frequency of all letters except space.

Input Format

Take a string as input

Output Format

Reverse Alphabetical order Characters followed by ‘\*’ sign to represents frequency of character

Sample Input 1

This is the final boarding call for passengers booked on flight AE to Kansas City

Sample Output 1

y:\*

t:\*\*\*\*

s:\*\*\*\*\*\*\*

r:\*\*\*

p:\*

o:\*\*\*\*\*\*

n:\*\*\*\*\*

l:\*\*\*\*

k:\*

i:\*\*\*\*\*\*

h:\*\*\*

g:\*\*\*

f:\*\*\*

e:\*\*\*\*

d:\*\*

c:\*

b:\*\*

a:\*\*\*\*\*\*

T:\*

K:\*

E:\*

C:\*

A:\*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case 1 | Test case 2 | Test case 3 | Test case 4 | Test case 5 |
| Chitkara University Java Class | Java is a high-level programming language originally developed by Sun Microsystems and released in 1995 | Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. | Java Coding Academy | Explore Paris holidays and discover the best time and places to visit. |
| y:\*  v:\*\*  t:\*\*  s:\*\*\*  r:\*\*  n:\*  l:\*  k:\*  i:\*\*\*  h:\*  e:\*  a:\*\*\*\*\*  U:\*  J:\*  C:\*\* | y:\*\*\*  v:\*\*\*  u:\*\*  t:\*  s:\*\*\*\*\*  r:\*\*\*\*\*  p:\*\*  o:\*\*\*\*  n:\*\*\*\*\*\*  m:\*\*\*  l:\*\*\*\*\*\*\*  i:\*\*\*\*\*\*\*  h:\*\*  g:\*\*\*\*\*\*  e:\*\*\*\*\*\*\*\*\*\*  d:\*\*\*\*  c:\*  b:\*  a:\*\*\*\*\*\*\*\*\*  S:\*  M:\*  J:\*  9:\*\*  5:\*  1:\*  -:\* | y:\*  w:\*  v:\*\*\*\*  u:\*\*\*  t:\*\*\*  s:\*\*\*\*\*\*\*\*  r:\*\*\*\*\*  p:\*  o:\*\*\*\*\*\*\*  n:\*\*\*\*\*  m:\*  l:\*  i:\*\*\*\*  h:\*\*  f:\*\*\*  e:\*\*\*  d:\*\*  c:\*\*  a:\*\*\*\*\*\*\*\*\*  X:\*  W:\*  U:\*  S:\*  O:\*  N:\*  M:\*  J:\*  I:\*  .:\*  ,:\*\*\* | y:\*  v:\*  o:\*  n:\*  m:\*  i:\*  g:\*  e:\*  d:\*\*  c:\*  a:\*\*\*  J:\*  C:\*  A:\* | y:\*  x:\*  v:\*\*  t:\*\*\*\*\*  s:\*\*\*\*\*\*  r:\*\*\*  p:\*\*  o:\*\*\*\*  n:\*\*  m:\*  l:\*\*\*  i:\*\*\*\*\*\*  h:\*\*  e:\*\*\*\*\*\*  d:\*\*\*\*  c:\*\*  b:\*  a:\*\*\*\*\*  P:\*  E:\*  .:\* |

*Solution:*

import java.util.\*;

public class SequenceLetter {

private static String sentence;

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

sentence=sc.nextLine();

TreeMap<Character,Integer>ts=new TreeMap<Character,Integer>();

ts=computeFrequency();

displayLetterFrequency(ts);

}

**static TreeMap<Character,Integer>computeFrequency() {**

**TreeMap<Character,Integer> tm = new TreeMap<Character,Integer>(new Comparator<Character>() {**

**public int compare(Character o1, Character o2) {**

**return o2.compareTo(o1);**

**}**

**});**

**char ch[]=sentence.toCharArray();**

**int n=ch.length;**

**Character c1;**

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

**int count=0;**

**c1=(Character)ch[i];**

**if(ch[i]!='0'&&ch[i]!=' ') {**

**for(int k=i;k<n;k++){**

**Character c2=(Character)ch[k];**

**if(c1==c2) {**

**count++;**

**ch[k]='0';**

**}**

**}**

**tm.put(c1,count);**

**}**

**}**

**return tm;**

**}**

**static void displayLetterFrequency(TreeMap<Character,Integer> frequencyMap) {**

**for(Map.Entry<Character,Integer>map:frequencyMap.entrySet()){**

**System.out.print(map.getKey()+":");**

**int n=(Integer)map.getValue();**

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

**System.out.print('\*');**

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

**}**

**}**

}