

Aditya Degree Colleges

Java Online Training Coding Test_4 Key

Date: 25-04-2020

Program - 1:

Given, N Mice and N holes are placed in a straight line. Each hole can accommodate only 1 mouse. A mouse can stay at his position, move one step right from x to $x + 1$, or move one step left from x to $x - 1$. Any of these moves consumes 1 minute. Write a program to assign mice to holes so that the time when the last mouse gets inside a hole is minimized.

Input:

First line of input contains a single integer T , which denotes the number of test cases.

T test cases follows, first line of each test case contains a single integer N which denotes the number of mice and holes.

Second line of each test case contains N space separated integers which denotes the position of mice initially.

Third line of each test case also contains N space separated integers which denotes the position of holes.

Output:

For each test case in a new line print the **minimum time** required in which all the mice can get into the holes.

Constraints:

$1 \leq T \leq 100$

$1 \leq N \leq 10000$

Example:

Input:

```
2
3
4 -4 2
4 0 5
8
-10 -79 -79 67 93 -85 -28 -94
-2 9 69 25 -31 23 50 78
```

Output:

```
4
102
```

Explanation:

Testcase1: The maximum absolute difference between mouse to the corresponding hole position.

Sample Input2:

Input:

4

1 2 3 6

-4 2 3 5

Output:

5

Source Code:

```
import java.util.*;
class Mouse_hole
{
    public static void main (String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int t=sc.nextInt();
        while(t-->0)
        {
            int n=sc.nextInt();
            int[] mice=new int[n];
            for(int i=0; i<n; i++)
                mice[i]=sc.nextInt();

            int[] hole=new int[n];
            for(int i=0; i<n; i++)
                hole[i]=sc.nextInt();

            Arrays.sort(hole);
            Arrays.sort(mice);

            int ma=0;
            for(int i=0; i<n; i++)
                ma=Math.max(ma, Math.abs(mice[i]-hole[i]));
            System.out.println(ma);
        }
    }
}
```

Program – 2:

Steve has a string of lowercase characters in range `ascii['a'..'z']`. He wants to reduce the string to its shortest length by doing a series of operations. In each operation he selects a pair of adjacent lowercase letters that match, and he deletes them. For instance, the string `aab` could be shortened to `b` in one operation. Steve's task is to delete as many characters as possible using this method and print the resulting string. If the final string is empty, print `Empty String`

Input Format

A single string, `s`.

Constraints

- $1 \leq \text{len}(s) \leq 100$

Output Format

If the final string is empty, print `Empty String`; otherwise, print the final non-reducible string.

Sample Test Case 1:

Input:

`aaabccddd`

Output

`abd`

Explanation

Steve performs the following sequence of operations to get the final string:

`aaabccddd` → `abccddd` → `abddd` → `abd`

Sample Test Case 2:

Input:

`aa`

Output

`Empty String`

Hidden Test Case 1:

Input:

`baab`

Output:

`Empty String`

Hidden Test Case 2:

Input:

`mwkommokwmxjivkkvijxshscbbcsghsgwdyqqydwgzpnlzzlnpvzfeaiiaefvyeqjccjqeymhqhiihqhmhaomkkmoahhddymmyddh`

Output:

`Empty String`

Hidden test Case 3:

Input:

`aabbccddeeffgghhiijjkkllmmnnoopppqqrrssttuuvvwwxxxyzz`

Output:

`Empty String`

Hidden Test Case 4:

Input:

acdqglrfkqyuqfjkxyqvnrtyzfrzrmzlygfveulqfpdbhlqdqrrqdqlhbdpfqluevfgylzmrzrfsytrnv
qyxkjqfquyqkfrlacdqj

Output:

acdqgacdqj

Source Code:

```
import java.util.Scanner;
public class Practice
{
    public static void main(String[] args)
    {
        Scanner scanner=new Scanner(System.in);
        String str=scanner.next();
        for(int i=1;i<str.length();i++)
        {
            if(str.charAt(i)==str.charAt(i-1))
            {
                str=str.substring(0,i-1)+str.substring(i+1);
                i=0;
            }
        }
        if(str.length()==0)
        {
            System.out.println("Empty String");
        }
        else
        {
            System.out.println(str);
        }
        scanner.close();
    }
}
```

MCQ_Day4_KEY:

Which of these is an incorrect Statement?

- A) Array can be initialized using comma separated expressions surrounded by square braces
- B) Array can be initialized when they are declared.
- C) It is necessary to use new operator to initialize an array.
- D) None of the above

ANSWER:C

Method overloading is differentiated

- A) By return type
- B) By method name
- C) By parameters data type
- D) All of the above

ANSWER: C

Which of these can be overloaded?

- A) Methods
- B) Constructors
- C) Both A and B mentioned
- D) None of the mentioned

ANSWER: C

What is the process of defining a method in terms of itself, that is a method that calls itself?

- A) Polymorphism
- B) Abstraction
- C) Encapsulation
- D) Recursion

ANSWER: D

```
class San
{
    public void m1 (int i,float f)
    {
        System.out.println(" int float method");
    }
    public void m1(float f,int i);
    {
        System.out.println("float int method");
    }
}
```

```

public static void main(String[]args)
{
    San s=new San();
    s.m1(20,20);
}
}

```

- A) int float method
- B) float int method
- C) compile time error
- D) run time error

ANSWER: C

What will be the output of the following Java code?

```

class overload
{
    int x;
    int y;
    void add(int a)
    {
        x = a + 1;
    }
    void add(int a, int b)
    {
        x = a + 2;
    }
}
class Overload_methods
{
    public static void main(String args[])
    {
        overload obj = new overload();
        int a = 0;
        obj.add(6);
        System.out.println(obj.x);
    }
}

```

- A) 5
- B) 6
- C) 7
- D) 8

ANSWER: C

```

class test
{
    int a;
    int b;
    void meth(int i , int j)
    {
        i *= 2;
        j /= 2;
    }
}
class Output
{
    public static void main(String args[])
    {
        test obj = new test();
        int a = 10;
        int b = 20;
        obj.meth(a , b);
        System.out.println(a + " " + b);
    }
}

```

- A) 10 20
- B) 20 10
- C) 20 40
- D) 40 20

ANSWER: A

What will be the output of the following Java program?

```

class equality
{
    int x;
    int y;
    boolean isequal()
    {
        return(x == y);
    }
}
class Output
{
    public static void main(String args[])
    {
        equality obj = new equality();
    }
}

```

```

        obj.x = 5;
        obj.y = 5;
        System.out.println(obj.isEqual());
    }
}

```

- A) false
- B) true
- C) 0
- D) 1

ANSWER: B

If the following methods are declared in a class, which of them fail to compile because of overloading rules?

```

int multiplication(int a, int b)    // 1
float multiplication(float a, int b) // 2
float multiplication(int a, float b) // 3
void multiplication(float a)       // 4
int multiplication(int a)          // 5
void multiplication(int a)         // 6

```

- A) 2, 3, 4, 5 and 6
- B) 4, 5 and 6
- C) 5 and 6
- D) All are correct

ANSWER: C

What is the value returned by function compareTo() if the invoking string is less than the string compared?

- A) zero
- B) value less than zero
- C) value greater than zero
- D) none of the mentioned

ANSWER: B

In the following Java code, which code fragment should be inserted at line 3 so that the output will be: "123abc 123abc"?

1. `StringBuilder sb1 = new StringBuilder("123");`
 2. `String s1 = "123";`
 3. `// insert code here`
 4. `System.out.println(sb1 + " " + s1);`
- A) `sb1.append("abc"); s1.append("abc");`
 - B) `sb1.append("abc"); s1.concat("abc");`
 - C) `sb1.concat("abc"); s1.append("abc");`

D) sb1.append("abc"); s1 = s1.concat("abc");

ANSWER: D

Which of these is used to access a member of class before object of that class is created?

A) public

B) private

C) static

D) protected

ANSWER: C

What will be the output of the following Java code?

class access

```
{
    public int x;
    private int y;
    void cal(int a, int b)
    {
        x = a + 1;
        y = b;
    }
}
public class access_specifier
{
    public static void main(String args[])
    {
        access obj = new access();
        obj.cal(2, 3);
        System.out.println(obj.x + " " + obj.y);
    }
}
```

A) 3 3

B) 2 3

C) Runtime Error

D) Compilation Error

ANSWER: D

Which of these access specifier must be used for class so that it can be access by sub class of other packages?

A) default

B) private

C) protected

D) none of the mentioned

ANSWER: C

What will be the output of the following Java code?

```
class output
{
    public static void main(String args[])
    {
        String chars[] = {"a", "b", "c", "a", "c"};
        for (int i = 0; i < chars.length; ++i)
            for (int j = i + 1; j < chars.length; ++j)
                if(chars[i].compareTo(chars[j]) == 0)
                    System.out.print(chars[j]);
    }
}
```

a) ab

b) bc

c) ca

d) ac

ANSWER: D

Which of these keywords can be used to prevent inheritance of a class?

A) super

B) constant

C) class

D) final

ANSWER: D

Private constructors are used in

A) Single tone class

B) Constructor Chain

C) Limiting creation of objects

D) All of the above

ANSWER:D

Constructor returns

A)void

B)int

C)float

D)Instance of class

ANSWER:D

Inheritance is useful for

- A) Acquiring properties of base class
- B) Acquiring properties of child class
- C) Acquiring properties of final class
- D) Acquiring properties of private class

ANSWER:A

Which of these is correct way of inheriting class A by class B?

- A) class B extends class A {}
- B) class B + class A {}
- C) class B inherits class A {}
- D) class B extends A {}

ANSWER:D