JP Morgan Chase & Co. Coding Questions

Question 1 : Copycat

Problem Statement :

Ashish was copying from Rahit in the exam. So, Rahit told him to change the answers a little bit so that the examiner cannot find the fraud. But silly Ashish in the way started to change all the answers that were needed. He shuffled the letters in each word in a way where the maximum number of letters were misplaced.

For a given word, find the maximum difference that Ashish can generate between his answer and Rahit's answer.

Suppose Rahit wrote "car" for an answer, Ashish can write "acr" with difference 2, or "arc" with differnece 3.

Note That: The letters are all in lowercase.

Input Format:

First line containing an integer n, number of words.

Then, n numbers of lines as the query words.

Output:

N number of lines with an integer each denoting possible maximum difference.

Sample Input:

4

abababa

bbj

kj

kk

Sample Output:

6

2

2

0

JAVA

```
import java.util.*;

class Main {

   public static String swapString(String s, int i, int j) {
```

```
    char[] b = s.toCharArray();
    char temp;
    temp = b[i];
    b[i] = b[j];
    b[j] = temp;
    return String.valueOf(b);
}

public static void generatePermutation(String s, int start, int end, HashSet < String > set) {

    if (start == end - 1)
        set.add(s + " ");
    else {
        for (int i = start; i < end; i++) {
            s = swapString(s, start, i);
            generatePermutation(s, start + 1, end, set);
            s = swapString(s, start, i);
        }
    }

}
public static int maxDiff(String str, String s) {
    int c = 0;
    for (int i = 0; i < s.length(); i++) {
        if (s.charAt(i) == str.charAt(i));
        else
            c++;
    }
    return c;

}
```

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

    String[] a = new String[n];
    for (int i = 0; i < n; i++) {
      a[i] = sc.next();
    }
    for (int i = 0; i < n; i++) {
      HashSet < String > set = new HashSet < String > ();
      String s = a[i];
      generatePermutation(s, 0, s.length(), set);
      int max = 0;
      int k = 0;
      for (String str: set) {

        k = maxDiff(str, s);
        max = Math.max(max, k);

      }
      System.out.println(max);
    }
  }

}
```

Question 2 :New Number System

Problem Statement :

Here is about to introduce a new kind of number system. Where instead of 10 digits there is 20, from a to t, all in small. Now a means 1, b means 2 and t means 20, thenn aa means 21. Now for a new number you have to print the decimal form it.

Note that the letters in the input string will be lower case and from a to t.

Input Format:

Single line containing the string of new number system's number

Output Format:

Single line denoting the integer with the same decimal value as the input string

Sample input 1: e

Sample Output: 5

Sample Input 2: ac

Sample Output: 23

JAVA

```
import java.util.*;
class Main {
   public static int func(String s, int n, int i) {
      if (n < 0)
         return 0;
      return (s.charAt(n) - 'a' + 1) * (int) Math.pow(20, i) + func(s, n - 1, i + 1);
   }
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String str = sc.next();
      System.out.println(func(str, str.length() - 1, 0));
   }
}
```

Question 3 : Seating Arrangement in Exam Hall

Problem Statement :

Semester exams are going on for university students. Examiners noticed that a group of people are trying to cheat. They marked students of that group as '1' and students of another group (who are not cheating) as '0'

We can reduce cheating by not allowing students from group 1 to sit together, means no two students from group 1 can sit together. Seatings are marked using above conditions. Your task is to give the seating placement of nth possibility Possibility order from 1 to 10 is given below

[1 10 100 101 1000 1001 1010 10000 10001 10010]

Sample input :

3 → number of test cases

4

6

9

Sample output :

101

1001

10001

Explanation :

4th possibility is 101

6th possibility is 1001

9th possibility is 10001

JAVA

```
import java.util.*;
class Main {
   public static void possibilities(int n) {
      int c = 0;
      String b = "";
      for (int i = 1; n != c; i++) {
         String s = Integer.toString(i, 2);
         if (!s.contains("11")) {
            c++;
            b = s;
         }
      }
      System.out.println(b);
   }

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

```
        int[] a = new int[tc];
        for (int i = 0; i < tc; i++) {
            a[i] = sc.nextInt();
        }
        for (int i = 0; i < tc; i++) {
            possibilities(a[i]);
        }
    }
}
```

Question 4 : Total Distinct Money

Problem Statement :

You woke up from sleep and found yourself in the 0th row and 0th column of a grid. every other square in a grid has some amount of money kept there. If you are given the matrix with all the values left in the cells, you have to find how many different ways are there to rich the r-1 th , c-1 th cell and the sum of all possible amount of money you will have each time if you bring all the money kept in places in the cell.

Note that, if you are in i,j th cell, either you can go i+1, j th cell or you can go i,j+1 cell.

Again, the 0,0th grid and the n-1,m-1 th grid will have 0 value.

Input Format:

Two integers R and C meaning the number of rows and columns.

Next R lines C space separated integers denoting the total grid.

Output Format:

First Line denoting the distinct ways to rich.

Next line denotes the total money if you use all possible distinct ways (Given that if you take the money from a cell, the money is readded in the cell).

Sample Input:

3 3

0 2 3

1 3 2

1 1 0

Sample Output:

4

21

Explanation:

The all possible totals are:

0 -> 2 -> 3 -> 2 -> 0 Total = 7

0 -> 2 -> 3 -> 2 -> 0 Total = 7

0 -> 2 -> 3 -> 1 -> 0 Total = 6

0 -> 1 -> 3 -> 2 -> 0 Total = 6

0 -> 1 -> 3 -> 1 -> 0 Total = 5

0 -> 1 -> 1 -> 1 -> 0 Total = 3

There are 4 distinct ways and the total is 21

JAVA

```
import java.util.*;
public class Main {
   static ArrayList < ArrayList < Integer >> a;
   static int n, m, t;
   static int ans = 0, sum = 0;
   static Map < Integer, Integer > mm = new HashMap < Integer, Integer > ();
   static void answer(int i, int j, int num, ArrayList < ArrayList < Integer >> a2) {
      if (i == n - 1 && j == m - 1) {
         if (mm.get(num) == null) {
            mm.put(num, 1);
            ans++;
            sum += num;
         }
         return;
      }
      if (i == n - 1) {
         answer(i, j + 1, num + a2.get(i).get(j), a2);
         return;
      }
      if (j == m - 1) {
         answer(i + 1, j, num + a2.get(i).get(j), a2);
```

```
            return;
        }
        answer(i + 1, j, num + a2.get(i).get(j), a2);
        answer(i, j + 1, num + a2.get(i).get(j), a2);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        m = sc.nextInt();
        ArrayList < ArrayList < Integer >> arr = new ArrayList < ArrayList < Integer >> ();
        for (int i = 0; i < n; i++) {
            ArrayList < Integer > temp = new ArrayList < Integer > ();
            for (int j = 0; j < m; j++) {
                temp.add(sc.nextInt());
            }
            arr.add(temp);
        }
        a = arr;
        answer(0, 0, 0, a);
        System.out.println(ans + "\n" + sum);
    }
}
```

Question 5 : Death Note

Problem Statement :

Ryuk, the Shinigami (God of death) had allowed Light Yagami, a school student, to kill as many people as he can by using a death note. But writing the names barely will allow other people to watch them. So he encrypts the names using digits, where a means 1, b means 2 and so on upto z is 26. Now if he gives numbers, there is a communication error because a number string can be decrypted by the death note in various ways and eventually killing them all. If everyone in the world has a unique name, for a given number, how many people can die?
NOTE THAT: There is every possible name in the world with the 26 letters, and capital or small letters is not a problem.

Input format:
A number stream denoting the first name's encrypted version

Output Format:
Number of people dying by this.

Constraints:
1<=stream length<=10^8

Sample Input: 1267
Sample Output: 3
Output Specification:Two people of the name azg and lfg die.

JAVA

```
import java.util.*;

class Main {
   static String s;
   static int ans;
   public static void func(int i, int n) {
      if (i == n - 1 || i == n) {
         ans++;
         return;
      }
      if (s.charAt(i) == '1')
         func(i + 2, n);
      else if (s.charAt(i) == '2' && s.charAt(i + 1) < '7')
         func(i + 2, n);
      func(i + 1, n);


   }
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      s = sc.next();
      func(0, s.length());
      System.out.println(ans);
   }
}
```

Question 6 : Choco and chocolate

Problem Statement :

Choco, a chocolate lover, has N amount of money with him. He wants to buy as much chocolate as possible. So, he goes to a chocolate shop “Bandyman ”. Mike, the owner of “Bandyman ” has different types of chocolate in his store (represented by a character) placed in a row.

Mike, give an offer to Choco that he can buy a selected type of chocolate for free and need to pay for the other types of chocolates and Choco can only buy consecutive chocolates.

Now, you need to write a code to find the maximum amount of chocolates Choco can get by selecting the chocolates optimally.

Input format :

1st line contains 2 space separated integers A and B denoting the number of chocolates and the amount of money Choco has.

The 2nd line contains A chocolates represented by a string. All chocolates represented by lowercase alphabets.

The 3rd line represents 26 space separated integers representing the cost to buy the chocolates. [First integer represents the cost of the chocolate of type ‘a’, 2nd integer represents the cost of the chocolates of type ‘b’ and so on]

Output format :

Print the maximum number of chocolates Choco can buy.

Constraints :
1<=A<=10^5
1<=B<=10^9
1<=cost of chocolate<=10^9

Sample input 1 :

6 10
aabcda
5 4 4 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Sample output 1 :
4

Explanation :

Choco can select the chocolate of type ‘a’ for free and start buying from index 0 and if he buys “aabc” then he has to pay less (0+0+4+4=8) than the total money he has.

This is the maximum number of chocolates he can get in this case.

JAVA

```
import java.util.*;


class Main {
```

```
public static int solve(int n, int amnt, String s, int[] price) {
    int[] freq = new int[26];
    char[] ch = s.toCharArray();
    int temp = 0, ans = 0, maxFreq = 0, st = 0;
    for (int i = 0; i < n; i++) { int indx = ch[i] - 'a'; freq[indx] += price[indx]; temp += price[indx];
maxFreq = Math.max(maxFreq, freq[indx]); if (temp - maxFreq > amnt) {
        while (temp > amnt) {
            boolean b = false;
            int tempIndx = ch[st] - 'a';
            if (maxFreq == freq[tempIndx]) {
                b = true;
            }
            temp -= price[tempIndx];
            freq[tempIndx] -= price[tempIndx];
            st++;
            if (b) {
                maxFreq = 0;
                for (int j = 0; j < 26; j++) {
                    maxFreq = Math.max(freq[j], maxFreq);
                }
            }
            if (temp - maxFreq < amnt) {
                break;
            }
        }
      }
      ans = Math.max(ans, i - st + 1);
    }
    return ans;
  }
  public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int amnt = sc.nextInt();
        String s = sc.next();
        int[] price = new int[26];
        for (int i = 0; i < n; i++) {
            price[i] = sc.nextInt();
        }
        System.out.println(solve(n, amnt, s, price));
    }
}
```

Question 7 :Number with 2

Problem Statement :

Suppose you are in a number system, where if the number doesn't contain 2 in the unit digit then the number is not valid. So the first number of the number system is 2, the second number is 12, and the third is 22.
for a given integer n, you have to print the nth element of the number system.

Input Format:
First line, containing n denoting the number of test cases.
then n number of lines for the query.

Output Format:
Print the consecutive number in the number system for each query.

Sample Input:
3

Sample Output:
22

Explanation:
1st number will be 2 , 2nd number will be 12 and third number will be 32

JAVA

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
```

```
        System.out.println((n - 1) * 10 + 2);
    }
}
```

Question 8 : Vowel Encryption

Problem Statement :

There is an encryption game going on. You will be given a number. If a digit is prime, it will take a vowel. Otherwise it will take a consonant value.
By this process, you have to make the string the lexicographically smallest possible. For a given number, print the output as a string.;

Input Format:
An integer n denoting the number.
Output Format:
The encrypted word.

Sample Input: 123421
Sample Output: baecab

JAVA

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        String vowel = "aeiou";
        char arr[] = new char[10];
        arr[2] = 'a';
        arr[3] = 'e';
        arr[5] = 'i';
        arr[7] = 'o';
        char ch = 'b';
        for (int i = 1; i < arr.length; i++) {
            if (vowel.indexOf(ch) != -1) {
                ch++;
                continue;
```

```
        } else if (arr[i] == 0)
            arr[i] = (char) ch++;
    }
    int temp = n;
    int res = 0;
    while (temp != 0) {
        res = res * 10 + temp % 10;
        temp = temp / 10;
    }
    int i = 1;
    while (res != 0) {
        System.out.print(arr[res % 10]);
        res = res / 10;
    }
  }

}
```

Question 9 :Jack's Text

Problem Statement :

Jack is learning to type english from the beginning and he is making an error of repeating the same words in his texts over whatsapp. Write a function that will take input for his text sent to you and then keep only the unique texts.
Note that, the uniqueness is about being word specific not position, there are nothing but alphabets in the sentences and words are separated only with white space.

Constraints:
Words in the line<=10^5
Alphabets in the words<=20

Sample Input:
Send send the image send to to to me

Output:
Send the mage to me

JAVA

```
import java.util.*;

class Main {
```

```
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String str = sc.nextLine();
    String arr[] = str.split(" ");
    String first = arr[0];
    arr[0] = arr[0].toLowerCase();
    LinkedHashMap < String, Boolean > map = new LinkedHashMap < String, Boolean > ();
    for (int i = 0; i < arr.length; i++)
      map.put(arr[i], true);
    Set s = map.entrySet();
    Iterator itr = s.iterator();
    System.out.print(first + " ");
    int j = 0;
    while (itr.hasNext()) {
      Map.Entry m = (Map.Entry) itr.next();
      if ((Boolean) m.getValue() == true && j == 1)
        System.out.print(m.getKey() + " ");
      j = 1;
    }
  }
}
```

Question 10 :A Good Prime Number

Problem Statement :

A prime number is a number which is divisible by one and itself. Also a number is called a good prime number if the sum of its digits is a prime number. For example a number 23 is a good prime number because the sum of 2 and 3 (2+3=5) is 5 which is a prime number. You are given an integer K. Your task is to find the kth good prime number that is greater than a provided number N.

For example , 232 is a good prime number since the sum of all digits is 7 which is a prime number whereas 235 is not a good prime number.

Input format :

- The first line contains an integer N.
- The next line contains an integer K.

Output format :

A single integer which is a Kth good prime number that is greater than a provided number N.

Constraints :

- 1<=N<=10^5
- 1<=K<<=10^5

Sample Input 1:

4 4

Sample Output 1:

12

Explanation :

Good prime numbers starting from 4 are 5,7,11(1+1=2 which is prime number),12(1+2=3 which is prime number),14(1+4=5 which is a prime number) and so on. Because the sum of digits of an individual number is a prime number And 4 th good prime number is 12 in this series.Hence the output is 12.

Sample Input 2:

17 5

Sample Output 2:

29

Explanation :

Good prime numbers starting from 17 are 20,21,23,25,29…and the 5th prime number is 29.Hence the output is 29.

JAVA

```
import java.util.*;

class Main {

  public static boolean isPrime(int n) {

    if (n <= 1)

      return false;

    if (n <= 3)

      return true;

    if (n % 2 == 0 || n % 3 == 0)

      return false;

    for (int i = 5; i * i <= n; i = i + 6)
```

```
        if ((n % i == 0) || (n % (i + 2) == 0))
            return false;
        return true;
    }
    public static int solve(int n, int k) {
        int c = 0;
        ArrayList < Integer > list = new ArrayList < > ();
        for (int i = n + 1; c < k; i++) {
            int temp = i;
            int sum = 0;
            while (temp != 0) {
                sum = sum + temp % 10;
                temp = temp / 10;
            }
            if (isPrime(sum)) {
                list.add(i);
                c++;
            }
        }
        return list.get(k - 1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int k = sc.nextInt();
        System.out.println(solve(n, k));
    }
}
```