

WIPRO PRP Logical Building

Questions Bank

1. Kaprekar Number

A Kaprekar number is a non-negative integer, the representation of whose square in that base can be split into two parts that add up to the original number again. For instance, 45 is a Kaprekar number, because $45^2 = 2025$ and $20+25 = 45$.

2. Harshad Number

Write a Program in Java to input a number and check whether it is a **Harshad Number** or **Niven Number** or not.

Harshad Number : In recreational mathematics, a Harshad number (or Niven number), is an integer (in base 10) that is divisible by the sum of its digits.

Let's understand the concept of Harshad Number through the following example:

The number 18 is a Harshad number in base 10, because the sum of the digits 1 and 8 is 9 ($1 + 8 = 9$), and 18 is divisible by 9 (since $18 \% 9 = 0$)

The number 1729 is a Harshad number in base 10, because the sum of the digits 1, 7, 2 and 9 is 19 ($1 + 7 + 2 + 9 = 19$), and 1729 is divisible by 19 ($1729 = 19 * 91$)

The number 19 is not a Harshad number in base 10, because the sum of the digits 1 and 9 is 10 ($1 + 9 = 10$), and 19 is not divisible by 10 (since $19 \% 10 = 9$)

The first few Harshad numbers in base 10 are:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 18, 20, 21, 24, 27, 30, 36, 40, 42, 45, 48, 50, 54, 60, 63, 70, 72, 80, 81, 84, 90, 100, 102, 108, 110, 111, 112, 114, 117, 120, 126, 132, 133, 135, 140, 144, 150, 152, 153, 156, 162, 171, 180, 190, 192, 195, 198, 200 etc.

3. Encoded Array

Maya has stored few confidential numbers in an array(array of int).to ensure that others do not find the numbers easily,she has applied asimple encoding.

ENCODING USED: Each array element has been substituted with a value that is the sum of its original value and its succeeding elements value.

i.e. $arr[i]=arr[i]+arr[i+1]$

e.g.value in $arr[0]=original\ value\ of\ arr[0]+original\ value\ of\ arr[1]$

also note that value of last element i.e. arr[last index] remains unchanged.

EXAMPLE-

It the original array-{2,5,1,7,9,3}

The encoded array would be-{7,6,8,16,12,3}

Provided the encoded array, you are expected to find the-

a. first number (value in index 0) in the original array

b. Sum of all numbers in the original array

the prototype of the function is:

```
public static void findOriginalFirstAndSum(int[] input);
```

when input1 is the encoded array.

The method is expected array.

The method is expected to-

- Find the value of the first number of the original array and store it in the member output1 and
- Find the sum of all numbers in the original array and store it in the output2

ASSUMPTION:

- The array elements can be positive and /or negative numbers

EXAMPLE 1:

Original array={2,5,1,7,9,3}

Encoded array={7,6,8,16,12,3}

First number in original array=2

Sum of all numbers in original array=27

Note:

Only the "Encoded array" will be supplied to the function and it is expected to do the processing to find expected result values

EXAMPLES 2:

Original array={8,-10,3,-15}

Encoded array={-2,-7,-12,-15}

First number in original array=8

Sum of all numbers in original array=-14

EXAMPLE 3:

Original array={3}

Encoded array={3}

First number in original array=3

Sum of all numbers in original array=3

EXAMPLE 4:

Original array={1,-14,8,23,0,0,-6,89,21,53}

Encoded array={-44,-37,31,23,0,-6,83,110,74,53}

First number in original array=1

Sum of all numbers in original array=144

4. USER ID GENERATION:

Joseph's team has been assigned the task of creating user-ids for all participants of an online gaming competition. Joseph has designed a process for generating the user-id using the participant's First_name, Last_Name, PIN code and a number N. The process defined by Joseph is as below-

STEP 1- Compare the lengths of FIRST_Name and Last_name of the participant. The one that is shorter will be called "Smaller Name" and the one that is longer will be called the "Longer name". If the both First_name and Last_name are of equal length, then the name that appears earlier in alphabetical order will be "Smaller Name" and the name that appears later in alphabetical order will be called the "Longer name".

STEP 2: The user-id should be generated as below-First Letter of the longer name+ Entire word of the smaller name+ digit at position N in the PIN when traversing the PIN from Left to right+ Digit at position N in the PIN when traversing the PIN from right to left.

STEP 3: Toggle the Alphabets of the user-id generated in step-2.i.e.upper-case alphabets should become lower-case and lower-case should become upper-case
Let us see few Examples.

EXAMPLE-1: If the participant's details are as below-

First_Name =Rajiv

LAst_Name =Roy

PIN=560037

N=6

STEP-1: Length of Last_name is less than the Length of First_name, so that Smaller name is "Roy" and the longer name is "Rajiv"

STEP-2: The user-id will be=First Letter of the longer name+Entire word of the smaller name+digit at position N in the PIN when traversing the PIN from left to right+Digit at position N in the PIN when traversing the PIN from right to left

= First Letter of "Rajiv" + Entire word of " Roy "+6th digit of PIN from left+6th digit of PIN from right.

=R+Roy+7+5

therefore,user-id=RRoy75

STEP-3: Toggle the alphabets in the user-id.So,user-id=rrOY75

EXAMPLE-2: If the participant's details are as below-

First_Name =Manoj

LAst_Name =Kumar

PIN=561327

N=2

STEP-1: Length of First_name is equal to the Length of Last_name .Alphabetically , "Kumar" appears earlier than "Manoj"(by comparing alphabetic positions of 'k' and 'M' so that Smaller name is "Kumar" and the longer name is "Manoj"

STEP-2:The user-id will be=First Letter of the longer name+Entire word of the smaller name+digit at position N in the PIN when traversing the PIN from left to right+Digit at position N in the PIN when traversing the PIN from right to left

= First Letter of "Manoj" + Entire word of " Kumar "+2th digit of PIN from left+2th digit of PIN from right.

=M+Kumar+6+2

therefore, user-id=MKumar62

STEP-3:Toggle the alphabets in the user-id.So,user-id=mkUMAR62

EXAMPLE-3: If the participant's details are as below-

First_Name =Kumud

LAst_Name =Kumar

PIN=561327

N=2

STEP-1: Length of First_name is equal to the Length of Last_name .Alphabetically , "Kumar" appears earlier than "Kumud"(by comparing alphabetic positions of 'A' and 'U' so that Smaller name is "Kumar" and the longer name is "KUMUD"

STEP-2:The user-id will be=First Letter of the longer name+ Entire word of the smaller name+ digit at position N in the PIN when traversing the PIN from left to right+ Digit at position N in the PIN when traversing the PIN from right to left

= First Letter of "Kumud" + Entire word of " Kumar "+2th digit of PIN from left+2th digit of PIN from right.

=K+Kumar+6+2

therefore, user-id=KKumar62

STEP-3:Toggle the alphabets in the user-id. So, user-id=kkUMAR62

You are part of Joseph's team and he has asked you to write a program(method) to generate the participants user-id using the above rules.

The prototype of the method(function) should be as below-

public static void useridgeneration(String input1, String input2,int input3,int input4)

where,

- is the First_Name
- is the Last_Name
- is the PIN
- is the number N

The method(function) should do the processing as per rules explained above and should assign the generated user-id to the member variable **Output1**

5. WifiPinGeneration

EXAMPLE 1

INPUT 1 – 210

INPUT 2 –KUMAR

OUTPUT – 2)KUMA1

EXAMPLE 2

INPUT 1 -105

INPUT 2 – BALA

OUTPUT – 6%BAL0

1	!
2	@
3	#
4	\$
5	%
6	^
7	&
8	*
9	(
0)

6. Sum of Power of Digits:

Alex has been asked by his teacher to do an assignment on powers of numbers. The assignment requires Alex to find the sum of powers of each digit of a given number, as per the method mentioned below.

If the given number is 582109, the sum of powers of digits will be calculated as=
=(5 raised to the power of 0)+(8 raised to the power of 5)+(2 raised to the power of 8)+(1 raised to the power of 2)+(0 raised to the power of 1)+(9 raised to the power of 0)|
i.e each digit of the number is raised to the power of the next digit on its left-side. Note that the left-most digit has to be raised to the power of 0. The sum of all of these powers is the expected result to be calculated.

Example 1:if the given number is 582109, the sum of power of digits=
=(5 raised to the power of 0)+(8 raised to the power of 5)+(2 raised to the power of 8)+(1 raised to the power of 2)+(0 raised to the power of 1)+(9 raised to the power of 0)|
=1+32768+256+1+0+1=33027

Example 2:if the given number is 12345, the sum of power of digits=

$$=(1 \text{ raised to the power of } 0)+(2 \text{ raised to the power of } 1)+(3 \text{ raised to the power of } 2)+(4 \text{ raised to the power of } 3)+(5 \text{ raised to the power of } 4)|$$
$$=1+2+9+64+625=701$$

Example 3:if the given number is 572, the sum of power of digits=

$$=(5 \text{ raised to the power of } 0)+(7 \text{ raised to the power of } 5)+(2 \text{ raised to the power of } 7)|$$
$$=1+16807+128=16936$$

Assumptions:For this assignment let us assume that the given number will always contains more than 1 digit. i.e. the given number will always be >9

7. Longest palindrome in a string

James and Johnny wanted to find the number of longest palindromes in a given string. They wanted to write a code for the following:

Step 1: separate the substrings of the given string and compare it with itself to check whether it is a palindrome.

Step 2: Find all the possible substrings in a given string.

Step 3:Compare the substrings and print the longest palindrome in the given string and count of the longest palindromes .

Input 1:"bananas"

Output:"anana"

Input 2:"abracadabra"

Output:"aca", "ada"

8. MOST FREQUENTLY OCCURRING DIGITS IN A SERIES OF NOS.

The task is to find the Most Frequently Occurring Digits in a series of input numbers and then derive a

key by adding all those digits. Below are a few examples.

EXAMPLE 1:

If the series of input numbers are [1237, 262, 666, 140]

We notice that,

0 occurs 1 time

1 occurs 2 times

2 occurs 3 times

3 occurs 1 time

4 occurs 1 time

5 occurs 0 time

6 occurs 4 times

7 occurs 1 time

8 occurs 0 times

9 occurs 0 times

We observe that

-4 is the highest frequency in this series and

-6 is the only digit that occurs 4 times.

Thus the most frequently Occuring digit in this series is only 6.

So, the key should be derived as= Sum of all Most Frequently Occurring digits=6

Note: if more than 1 digit occurs the most frequent time, key is obtained by adding all the digits as given below.

EXAMPLE 2:

If the series of input numbers are [1237, 202, 666, 140]

We notice that,

0 occurs 2 times

1 occurs 2 times

2 occurs 3 times

3 occurs 1 time

4 occurs 1 time

5 occurs 0 time

6 occurs 3 times

7 occurs 1 time

8 occurs 0 times

9 occurs 0 times

We observe that

-3 is the highest frequency in this series and

- 2 and 6 are digits that occurs 3 times.

So, the key should be derived as= Sum of all Most Frequently Occurring digits= 2+6=8

EXAMPLE 3:

If the series of input numbers are [1237, 202, 666, 148, 88]

We notice that,

0 occurs 1 time

1 occurs 2 times

2 occurs 3 times

3 occurs 1 time

4 occurs 1 time

5 occurs 0 time

6 occurs 3 times

7 occurs 1 time

8 occurs 3 times

9 occurs 0 times

We observe that

-3 is the highest frequency in this series and

- 2,8 and 6 are digits that occurs 3 times.

So, the key should be derived as= Sum of all Most Frequently Occurring digits=
2+6+8=16

The rules for finding the key is as above rules.

Help on usage of ArrayList and Integer:

Useful ArrayList methods::

1. `size()` returns the number of elements in the ArrayList.

for ex:

`input1.size();` will return the number of Integer elements in input1

2. `get(int n)` fetches the nth element in the ArrayList

for ex:

`input1.get(n);` will return the nth Integer element in input1

Useful Integer methods:

1. `intValue();` returns the 'int' value from the integer

for ex:

`input1.get(n).intValue();` will return the int value from the nth integer element in input1

9. Encoding Three Strings (John, Johnny):

Anand was assigned the task of coming up with an encoding mechanism for any given three strings. He has come up with the below plan. STEP ONE: Given any three strings. break each string into 3 parts each.

For example - if the three strings as below

`input1="john"`

`input2="johny"`

`input3="janardhan"`

"john" should be split into "j" , "oh", "n" as FRONT,MIDDLE and END parts respectively.

"johny" should be split into "jo","h","ny" as the FRONT,MIDDLE and END parts respectively.

"janardhan" should be split into "jan","ard","han" as the FRONT,MIDDLE and END parts respectively.

i.e if the no. of characters in the string are in multiples of 3 then each split-part will contain equal no.of characters as seen in the example of "janardhan"

If the no. of characters in the string are NOT in multiple of 3, and if there is one character more than multiple 3, then the middle part will get the extra characters, as seen in the example of "John".

If the no. of characters in the string are NOT in multiples of 3, and if there are two characters more than multiple of 3, then the FRONT and END parts will get one extra character each, as seen in the example of "Johnny".

STEP TWO: Concatenate (join) the FRONT , MIDDLE and END parts of the strings as per the below specified concatenation-rule to form three Output strings.

Output1 = FRONT part of input1 + MIDDLE part of input2 + End part of input3

Output2 = MIDDLE part of input2 + END part of input2 + FRONT part of input3

Output3= END part of input1 + FRONT part of input2 + MIDDLE part of input3

For example, for above specified example input strings,

Output1 = "J" + "h" + "han"= "Jhhan"

Output2= "oh"+"ny"+"Jan"="ohnyJan"

Output3= "n"+"Jo"+"ard"="nJoard"

Step THREE : Process the resulting output strings based on the output-processing rule.

After the above two steps, we will now have three output strings. Further processing is required only for the third output string as per below rule -

"Toggle the case of each character in the string", i.e. in the third output string, all lower-case characters should be made upper-case and vice versa.

For example, for the above example strings, Output3 is "nJoard", so after applying the toggle rule, Output3 should become "NjOARD".

Final Result - The three output strings after applying the above three steps is the final result. i.e. for the above example,

Output 1 = "Jhhan"

Output2= " ohnyJan"

Output= "NjOARD"

Anand approaches you to help him write a program that would do the above mentioned processing on any given three strings and generate the resulting three output strings.

The prototype of the method(function) should be as below

public static void encode Three Stings(String input1, String input2, String input3)

where, input1,input2 and input3 are the three given input strings.

The method (function) should do the processing on these three strings and assign the resultant strings to member variables output1, output2 and output3.

Example 1:

FRONT,MIDDLE and END parts respectively.

Input1="Raagam"

Input2= "Talam"

Input3="Pallavi"

Output1="Ralvi"

Output2= "agamPa"

Output3= "AMtALLA"

10. Reversing Words with options:

Write a function (method) that takes as input a string (sentence) and generates a new string (modified sentence) by reversing the words in the original string, maintaining the words positions.

In addition, the function should be able to control the reversing of the case (uppercase or lowercase) based on a `case_option` parameter, as follows –

If `case_option = 0`, Normal reversal of words

i.e. If the original sentence is “Wipro TechNologies BangaLore”,
the new reversed sentence should be “orpiW seigoloNhceT eroLagnaB”

If `case_option = 1`, Reversal of words with retaining position’s Case

i.e. If the original sentence is “Wipro TechNologies BangaLore”,
the new reversed sentence should be “Orpiw SeigOlonhcet ErolaGnab”

Note that positions 1, 7, 11, 20 and 25 in the original string are uppercase W, T and B.
Similarly positions 1, 7, 11, 20 and 25 in the new string are uppercase O, S, O, E and G.

If `case_option = 2`, Reversal of words including Case

i.e. If the original sentence is “Wipro Technologies Bangalore”,
the new reversed sentence should be “ORPIw SEIGOLONHCeT EROLAGNAb”

Note that W, T and B, that were uppercase in original string have become lowercase and all the lowercase letters have become uppercase.

The prototype of the function is:

public static void reverseWords_andCase(String sentence, int case_option);

The reversed string is expected to be assigned to the member **String output1**.

NOTE:

- a) Only space character should be treated as the word separator. i.e. “Hello World” should be treated as two separate words “Hello” and “World”. However, “Hello,World”, “Hello;World”, “Hello-World” or “Hello/World” should be considered as a single word.
- b) Non-alphabetic characters in the string will not be subjected to case-changes. For example, if `case_option=1`, and the original sentence is “Wipro TechNologies, Bangalore”, the new reversed sentence should be “Orpiw ,seiGolonhcet Erolagnab”. Note that comma has been treated as part of the word “Technologies,” and because comma had to take the position of uppercase T it remained as a comma and uppercase T took the position of comma. However the words “Wipro” and “Bangalore” have changed to “Orpiw” and “Erolagnab”.

c) Some more examples –

Input string = “I Am alWays 24#7 Busy.”

With case-option = 0, reversed string will be “I mA syaWla 7#42 .ysuB”

With case-option = 1, reversed string will be “I Ma syAwla 7#42 .ysuB”

With case-option = 2, reversed string will be “i Ma SYAwLA 7#42 .YSub”

11. Least Frequent Digit

You need to find which digit occurs least number of times across the four given input numbers.

The prototype of the function is:

public static void LeastFrequentDigit (int input1, int input2, int input3, int input4);

where,

input1, input2, input3 and input4 are the four given input numbers. The function is expected to find the least frequent digit and place the result in the class variable output1.

Example1 --

If input1 = 123, input2 = 234, input3 = 345, input4 = 453

We see that across these four numbers,

1 occurs once,

2 and 5 occurs twice,

4 occurs thrice, and

3 occurs four times.

Therefore, 1 is the least frequent digit and hence the **output1=1**

NOTE - If more than a digit occurs the same number of least times, then the lowest (minimum) of those digits should be the result. Below example illustrates this.

Example2 -

If input1 = 123, input2 = 456, input3 = 789, input4 = 1287

We see that

3, 4, 5, 6 and 9 occurs once, and,

1, 2, 8 and 7 occurs twice,

As there are five digits (3, 4, 5, 6 and 9) that occurs least number of times, the result will be the lowest (min) digit out of these five. Hence, **output1 = 3**

Let us see couple of more examples -

Example3 -

If input1 =1203, input2 = 7624, input3 = 2046, and input4 = 1052

The digit 3, 5 and 7 occurs once, and 3 is the lowest of these.

Therefore, the least frequent digit, **output1 = 3**

Example4 -

If input1 = 1205, input2 = 7624, input3 = 2046, and input4 = 1002

The digits 5 and 7 occur once, and 5 is lower of these.

Therefore, the least frequent digit, **output1 = 5**

12. Decreasing Sequence:

Given an integer array, find the number of decreasing sequences in the array and the length of its longest decreasing sequence.

The prototype of the function is as below:

public static void decreasingSeq(int[]input1);

The function takes as input an integer array **input1**.

The function sets the **output1** variable to the number of decreasing sequences in the array, and sets the **output2** variable to the length of the longest decreasing sequence in the array.

Example 1:

input1[]={11,4,3,1,4,7,8,12,2,3,7}

output1 should be 2

output2 should be 4

Explanation:

In the given array **input1** the decreasing sequences are "11,4,3,1" and "12,2" i.e, there are **two** decreasing sequences in the array, and so **output1** is assigned **2**. The first sequence in the array, and so **output1** is assigned **2**. The first sequence i.e, "11,4,3,1" is the longer one containing **four** items. When compared to the second sequence "12,2" which contains 2 items. So, the length of the longest decreasing sequence **output2=4**.

Example 2:

input1[]={1,3,2,1}

output1 should be 1

output2 should be 3

Explanation:

In the given array **input1** there is only one decreasing sequence which is "3,2,1" containing three items. So the number of decreasing sequences **output1** is assigned the value **1**. The length of the longest decreasing sequence **output2** is assigned the value **3**.

Example 3:

input1[]={12,15,21,29,69,79}

output1 should be 0

output2 should be 0

Explanation:

In the given array **input1** there is NO decreasing sequence. All the items are in increasing order, hence the number of decreasing sequences **output1** is assigned **0**. Similarly, the length of the longest decreasing sequence is also assigned the value **0**.

13. Password Generation

You should generate password by using input1,input2 with the following conditions.

Inputs:

Input1 - must be a string with any length

Input2 - must be a number

Output

You should generate 4 digit password based on the following requirements.

Unit Digit of Password = Last Character of input1

TenthDigit of Password = Digitsum of input2

100th Digit of password = First Character of input1.If its character is an UpperCase, needs to convert it into lowercase.If its character is a lowercase, need to convert into uppercase.

1000th Digit of password = 100th Digit of input2

Example:

Input1="Welcome"

Input2=129

Unit Digit of Password = e (Last Character of input1 - "Welcome")

Tenth Digit of password=3 (Digit sum of 129=1+2+9=12=1+2=3)

100th Digit of password=w(First character of Welcome is 'W'.Since it is uppercase we have to convert it into lowercase).

1000th Digit of password=1(100th Digit of 129 is 1).

Final output will be : **1w3e.**

14. Perfect Number

Write a program to check whether the given number is perfect number or not. If it is a Perfect Number It Should Print "YES" else It should print "NO".

Perfect Numbers are,

IF you add the factors of the number , you will be getting the same number as the output is known as Perfect Number.

Ex:

1. Input1=6

Factors of 6 is (1,2,3),So the sum of those factors are $1+2+3=6$. So 6 is Perfect Number.

2. Input1=4

Factors of 4 is(1,2), So the sum of those factors are $1+2=3$, So 4 is not a perfect number.

15. Word Length

Write a java program to find the length of each word in given sentence and concatenate or join those words length

Note: if the length of each word greater than or equal to 10 and use **digit sum technique**

(i.e. the word length is 15 = $1+5=6$)

Example:

input1: welcome to wipro

output1:725

input2:wipro technologies

output2:53

input3:talentnext

outpu3:1

16. World Wide Web

Crazy Zak has designed the below steps which can be applied on any given string (sentence) to produce a number.

STEP1. In each word. find the Sum of the Difference between the first letter and the last letter. second letter and the penultimate letter, and so on till the center of the word.

STEP2. Concatenate the sums of each word to form the result.

For example —

If the given string is "WORLD WIDE WEB"

STEP1. In each word, find the Sum of the Difference between the first letter and the last letter, second letter and the penultimate letter, and so on till the center of the word.

WORLD = [W-D]-40-L],[R] = [23-4]+[15-12]+[18] = [19],[3]+[18] = [40]

WIDE = [W-E]+[1-D] = [23-5]+[9-4] = [18],[5] = [23]

WEB = [W-B]+[E] = [23-2]+[5] [21]+[5] = [26]

17. Generate Wi-Fi pin for Hyatt group of Hotels:

The Hyatt Group of Hotels have approached you to automate the process of generating the Wi-Fi pin which will be used by guests during their stay at Hyatt.

The Wi-Fi pin must be a combination of room number, the guest's last name and one special character. The room numbers will always have three digits, i.e. it starts from 100 and can span up to 999.

The Wi-Fi pin is always a string of length 4, a combination which is created as per below rules -

Rule 1 - The unit's value of the Wi-Fi pin must be same as the tens value of the room number

Rule 2 - The tens value of the Wi-Fi pin must be same as the last character of the guests' last name.

Rule 3 - The hundreds value of the Wi-Fi pin must the special character equivalent of the units value of the room number. The special character details to be used are as given in table below

—

1	!
2	@
3	#
4	\$
5	%
6	^

7	&
8	*
9	(
0)

Rule 4 - The thousands value of the Wi-Fi pin must be created by performing digit sum on the room number i.e. by performing continuous addition of all the digits of the room number till we get a single digit number. If the obtained number is an even number, then it is used as it is. If the obtained single digit number is an odd number, then it is decremented by 1 to get the previous even number and is used. In Rule 4, zero should be considered as a neutral number (neither even nor odd). So, if sum of all digits results to 1 then the even number 2 must be used

18.Summet Sum:

Sum of largest 3 digit number from given 5 numbers

Given input numbers, sumeet has to find the sum of largest number that can be produced using 3 digits from each of the above 5 numbers

Example 1:

23792, 37221, 10270, 73391, 12005

number produced are 973, 732, 721, 973, 521 and sum is 3920

Example 2:

26674, 105, 37493, 95278, 27845

number produced are 766, 510, 974, 987, 875 and sum is 4112

Sum of smallest 2 digit number from given 5 numbers

Given input numbers, sumeet has to find the sum of smallest number that can be produced using 2 digits from each of the above 5 numbers

Example 1:

23792, 37221, 10270, 73391, 12005

number produced are 22,12,00,13,00 and sum is 47

Example 2:

26674,105,37493,95278,27845

number produced are 24,01,33,25,24 and sum is 107

19.Sum of Prime Index:

Given the array of N elements, find the sum of values that are present in **prime index** of the given array.

Example 1:

array { 10,20,30,40,50,60,70,80,90,100}

values at prime index are 30,40,60,80 and sum is 120.

Example 2:

array { -1,-2,-3,3,4,-7}

values at prime index are -3,3,-7 and sum is -7.

Example 3:

array { -4,5}

sum is 0.

Given the array of N elements, find the sum of values that are present in **non prime index** of the given array.

Example 1:

array { 10,20,30,40,50,60,70,80,90,100}

values at prime index are 10,20,50,70,90,100 and sum 340.

Example 2:

array { -1,-2,-3,3,4,-7}

values at prime index are -1,-2,4 and sum is 1.

Example 3:

array { -4,5}

sum is 1.

20.String Distance Replacement:

Encrypting word by String Distance Replacement

Eg: board -----> bqopasrvd

The sum of first two adjacent characters are 'b' and 'o' are 17 ($b=2+ o=15$), 17 corresponds to 'q'. so insert 'q' in between 'b' and 'o'. similarly same condition to all.

$o+a=16 \rightarrow 'p'$

$a+r=19 \rightarrow 's'$

$r+d=22 \rightarrow 'v'$

Special Cases:

1. if sum of two char exceed 26, apply modulus 26 and replace with result

eg: $l+o=27 \rightarrow 27\%26=1 \rightarrow 1$

2. if modulus 26 value is ==0 replace with '0' between character

eg: $z+z=52 \rightarrow 52\%26=0 \rightarrow '0'$

3. Spaces should be retained

4. if any character preceded or followed by a digit or non character just put it as it is

eg: 5c or c5 ----> 5c,c5

inut : wipro technologies

output : wfiyphrgo tyehckhvncoalaovginexs