**Accenture Coding Questions**

The following is a compilation ofthe type of **Accenture coding questions**that you’d be encountering.

**1. Execute the given function.**

def differenceofSum(n.m)

The function takes two integrals m and n as arguments. You are required to obtain the total of all integers ranging between 1 to n (both inclusive) which are not divisible by m. You must also return the distinction between the sum of integers not divisible by m with the sum of integers divisible by m.

Assumption

* m > 0 and n > 0
* Sum lies within the integral range

**Example**

Input:  
m = 6  
n = 30

Output:  
285

* Integers divisible by 6 are 6, 12, 18, 24, and 30. Their sum is 90.
* Integers that are not divisible by 6 are 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 25, 26, 27, 28, and 29. Their sum is 375.
* The difference between them is 285 (375 – 90).

Sample input:  
m = 3  
n = 10

Sample output:  
19

**2. Execute the given function.**

def LargeSmallSum(arr)

The function takes an integral arr which is of the size or length of its arguments. Return the sum of the second smallest element at odd position ‘arr’ and the second largest element at the even position.

**Assumption**

* Every array element is unique.
* Array is 0 indexed.

**Note:**

* If the array is empty, return 0.
* If array length is 3 or <3, return 0.

Example

Input:  
Arr: 3 2 1 7 5 4

Output:  
7

**Explanation**

* The second largest element at the even position is 3.
* The second smallest element at the odd position is 4.
* The output is 7 (3 + 4).

**3. Write a function to validate if the provided two strings are anagrams or not. If the two strings are anagrams, then return ‘yes’. Otherwise, return ‘no’.**

Input:

Input 1: 1st string  
Input 2: 2nd string

Output:  
(If they are anagrams, the function will return ‘yes’. Otherwise, it will return ‘no’.)

Example

Input 1: Listen  
Input 2: Silent

Output:  
Yes

Explanation

Listen and Silent are anagrams (an anagram is a word formed by rearranging the letters of the other word).

**Also Read About**- [Difference between argument and parameter](https://www.codingninjas.com/studio/library/difference-between-argument-and-parameter)

**Accenture Coding Questions in Python**

**4. Perform the following function.**

def Productsmallpair(sum,arr)

This function accepts the integers sum and arr. It is used to find the arr(j) and arr(k), where k ! = j. arr(j) and arr(k) should be the smallest elements in the array.

Keep this in mind:

* If n<2 or empty, return –1.
* If these pairs are not found, then return to zero.
* Make sure all the values are within the integer range.

Example

Input:  
Sum: 9  
Arr: 5 4 2 3 9 1 7

Output:  
2

**Explanation**

From the array of integers, we have to select the two smallest numbers, i.e 2 and 1. Sum of these two (2 + 1) = 3. This is less than 9 (3 < 9). The product of these two is 2 (2 x 1 = 2) Hence the output is integer 2.

Sample input:  
Sum: 4  
Arr: 9 8 –7 3 9 3

Sample output:  
–21

**5. Perform the function for the given purpose.**

For writing numbers, there is a system called N-base notation. This system uses only N-based symbols. It uses symbols that are listed as the first n symbols. Decimal and n-based notations are 0:0, 1:1, 2:2, …, 10:A, 11:B, …, 35:Z.

Perform the function: Chats DectoNBase(int n, int num)

This function only uses positive integers. Use a positive integer n and num to find out the n-base that is equal to num.

**Steps**

* Select a decimal number and divide it by n. Consider this as an integer division.
* Denote the remainder as n-based notation.
* Again divide the quotient by n.
* Repeat the above steps until you get a 0 remainder.
* The remainders from last to first are the n-base values.

Assumption  
1 < n < = 36

Example

Input:  
N: 12  
Num: 718

Output:  
4BA

**Explanation**

|  |  |  |  |
| --- | --- | --- | --- |
| num | Divisor | Quotient | Remainder |
| 718 | 12 | 59 | 10(A) |
| 59 | 2 | 4 | 11(B) |
| 4 | 12 | 0 | 4(4) |

Sample input:  
N: 21  
Num: 5678

Sample output:  
CI8

**6. Execute the function for the given purpose.**

When the sum of the digits exceeds a total of 9, a carry digit is added to the right-left of the digit. Execute the function: Int Numberofcarry(Integer num 1, Integer num 2)

**Assumption**

num1, num2 > = 0

Example

Input:  
num1: 451  
num2: 349

Output:  
2

**Explanation**

When we add these two numbers from the right to the left, we get two carries. The value of each of the carries is 1. Hence, the output is the total of these two carries, i.e., 2.

Sample input:  
num1: 23  
num2: 563

Sample output:  
0

**7. The given function has a string (str) and two characters, ch1 and ch2. Execute the function in such a way that str returns to its original string, and all the events in ch1 are replaced by ch2, and vice versa.**

Replacecharacter(Char str1, Char ch1, Int 1, Char ch2)

**Assumption**

This string has only alphabets that are in lower case.

Example

Input:  
str: apples  
ch1: a  
ch2: p

Output:  
paales

Explanation  
All the ‘a’ in the string is replaced with ‘p’. And all the ‘p’s are replaced with ‘a’.

**Accenture Coding Questions In Java**

**8. Perform the function: Int operationchoices(int c, int n, int a, int b). This function considers three positive inputs of a, b and c.**

Execute the function to get:  
(a + b), if c = 1  
(a / b), if c = 4  
(a – b), if c = 2  
(a x b), if c = 3

Example:

Input:  
a: 12  
b: 16  
c: 1

Output:  
28

Explanation  
C = 1, hence the function is (a + b). Hence, the output is 28.

Sample input:  
a: 16  
b: 20  
c: 2

Sample output:  
–4

**9. Perform the function Int calculate(int m, int n). This function needs two positive integers. Calculate the sum of numbers between these two numbers that are divisible by 3 and 5.**

Assumption  
m > n > = 0

Example

Input:  
m: 12  
n: 50

Output:  
90

Explanation  
The numbers between 12 and 50 that are divisible by 3 and 5 are 15, 30, and 45. The sum of these numbers is 90.

Sample input:  
m: 100  
n: 160

Sample output:  
405

**10. Execute the function for the given purpose.**

Create a matrix and mention the elements in it. Now, divide the main matrix into two halves in such a way that the elements in index 0 are even, the elements in index 1 are odd, and so on.

Then arrange the values in ascending order for even and odd. After this, calculate the sum of the second largest numbers from both even and odd matrices.

Example  
The size of the array is 5.  
Element at 0 index: 3  
Element at 1 index: 4  
Element at 2 index: 1  
Element at 3 index: 7  
Element at 4 index: 9

Even array: 1,3,9  
Odd array: 4,7

**Accenture Coding Questions in C**

**11. The binary number system only uses two digits 1 and 0.**

Perform the function: Int OperationsBinarystring(char\* str)

**Assumptions**

* Return to –1 if str is null.
* The str is odd.

Example:

Input:  
Str: ICOCICIAOBI

Output:  
1

Explanation  
The input when expanded is “1 XOR 0 XOR 1 XOR 1 XOR 1 AND 0 OR 1”. The result becomes 1 and hence the output is 1.

**12. Perform the function Checkpassword (char str[], int n)**

Execute the function which happens to be 1 if the str is a valid password or it remains 0.

Conditions for a valid password:

* The password should have at least 4 characters.
* It should have at least 1 digit.
* It should have one capital letter.
* It should not have any spaces or obliques (/).
* The first character should not be a number.

Assumption  
The input is not empty.

Example

Input:  
aA1\_67

Output:  
1

**13. Execute this function Void MaxInArray(int arr[], int length)**

This function helps in finding the maximum element in the array. Execute this function to print the maximum element in the array with its index.

**Assumptions**

* The index in the array for all the elements starts at 0.
* The maximum element is in a different line in the output.
* There has to be only one maximum element.
* The function prints only what is required.

Example

Input:  
23 45 82 27 66 12 78 13 71 86

Output:  
86  
9

Explanation  
The maximum element is 86 and the array is 9.

**14. Change frequent character**

The function to execute is  
Chat*Frequentcharacter(Char* str, char x)

This function has a string and a character. This function requires replacing the most used character in the str with the ‘x’ character.

Note: If two characters have the same frequency, then we will have to consider the frequency which has the lower ASCII value.

Example

Input:  
str: bbadbbababb  
char x: t

Output:  
ttadttatatt

Explanation  
The maximum character repeated is ‘b’ that is replaced with ‘t’.  
 Also read, [Software Testing](https://www.codingninjas.com/studio/library/introduction-to-software-testing)

**15. Execute the function Def Autocount(n).**

The function accepts the string n. It checks whether the number is an autobiographical number or not. If an integer returns, then it is an autobiographical number. If 0 returns, then it is not an autobiographical number.

**Assumption**

* The input value should not be more than 10 characters.
* The input string will have numeric characters.

Example

Input:  
N: “1210”

Output:  
3

Explanation  
The 0th position has the number 1, the 1st position has the number 2, the 2nd position has the number 1, and the 3rd position has number 0. Hence, it is an autobiographical number.

The count of autobiographical numbers in the input is 3, hence 3 is returned.  
   
Click here to learn about, [Html interview questions](https://www.codingninjas.com/studio/library/html-interview-questions-4065)

**16. In a given list of integers, find the number that has the highest frequency. If there are one or more such numbers, give the smaller one as output.**

Input:  
3  
7  
2 4 5 2 3 2 4  
6  
1 2 1 1 2 1  
10  
1 1 1 1 1 1 1 1 1 1

Output:  
2  
1  
1

**17. Execute the function for the given purpose.**

Write a function mergeArrays which merges two sorted arrays to create one single sorted array. Complete the function int\* mergeArrays(int a[], int b[], int asize, int bsize) below which takes the pointers to the first element of the two sorted arrays and the size of the arrays, and returns the base address of the final sorted array.

Input:  
4 // Size of 1st array  
1 2 3 6 // Elements of 1st array  
3 // Size of 2nd array  
4 5 7 // Elements of 2nd array

Output:  
1  
2  
3  
4  
5  
6  
7

**18. Create web access management to the kth largest number. It will accept an integer k and an array arr as its conditions and has to return the greatest element based on the value of k. That is, if k = 0, return the greatest element. If k = 1, return the second greatest element, and so on.**

**Example**

Suppose the array contains values like {74, 85, 102, 99, 101, 56, 84} and the integer k is 2. The method will return 99, the third greatest element, as there are only two (according to the value of k) values greater than 99 (101 and 102).

**19. We have mentioned a list of integers that have no duplicates. Find how many swaps it will take to sort the list in ascending order using Bubble sort.**

Input:  
3  
5  
2 1 4 6 3  
10  
123 21 34 45 25 675 23 44 55 900  
1  
23

Output:  
3  
16  
0

**20. Write a program to count the number of swaps required to sort a given list of integers in ascending order using the selection sort algorithm.**

Input:  
2  
3  
4 2 5  
5  
10 11 8 7 1

Output:  
1  
3

**21. Form an array of 1000 integers and find out the second-largest number. If there is no second largest number, return the value to –1.**

Example  
Input 1:  
3  
Input 2:  
{2,1,2}  
Output:  
1  
Explanation  
The integer 1 is the second largest in the array.  
Example  
Input 1:  
5  
Input 2:  
{4,7,9,8,0}  
Output:  
8

**22. Adam decides to do some charity work. From day 1 till day n, he will give i^2 coins to charity. On day ‘i’ (1 < = i < = n), find the number of coins he gives to charity.**

Example 1  
Input:  
2  
Output:  
5  
Explanation:  
There are 2 days.  
Example 2

Input:  
3

Output:  
14

**23. Perform a function to reverse a string word-wise. The input here will be the string. In the output, the last word mentioned should come as the first word and vice versa.**

Example  
Input:  
Welcome to code  
Output:  
code to Welcome

Explanation  
The Reversed string word wise function is applied.  
Example  
Input:  
Code to Crack Puzzle  
Output:  
Puzzle Crack to Code

**24. Find the sum of the divisors for the N integer number.**

Example 1  
Input:  
6  
Output:  
12

Explanation  
Divisors of 6 are 1, 2, 3, and 6. The sum of these numbers is 12.

Example 2  
Input:  
36  
Output:  
91

**25. Execute a function that accepts the integer array of length ‘size’ and finds out the maximum number that can be formed by permutation.**

Note: You will have to rearrange the numbers to make the maximum number.

Example  
Input:  
34 79 58 64  
Output:  
98765443

Explanation  
All digits of the array are 3, 4, 7, 9, 5, 8, 6, and 4. The maximum number found after rearranging all the digits is 98765443.

**26. Find a string of a length of 1000 for a large number. Output is the modulo of 11. The output specification is to return the remainder modulo 11 of the input.**

Input:  
121  
Output:  
0

Explanation  
121 mod 11 = 0

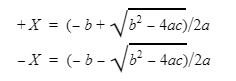
**27. Find out if the given set of points are on a straight line or not. If the points are on a straight line, then return the equation. If not, then return 0.**

Example  
Input:  
3  
1 1 2 2 3 3  
Output:  
1x – 1y = 0

Explanation  
The three points here are [1,1], [2,2], and [3,3]. These lie on a line, so the function returned the equation.

**28. Write a function to find roots of a quadratic equation ax^2 + bx + c = 0.**

Note: The formula to find the roots of a quadratic equation is given below:



Example  
Input 1: 1  
Input 2: –2  
Input 3: 3  
Output:  
{3.0,–1.0}

Explanation:  
On substituting the values of a, b, and c in the formula, the roots will be as follows:  
+X = 3.0  
-X = –1.0

**29. Write a function to find if the given string is a palindrome or not. Return 1 if the input string is a palindrome, else return 0.**

Input:  
level  
Output:  
1

Explanation:  
The reverse of the string ‘level’ is ‘level’. As they are the same, the string is a palindrome.

**30. Write a function to check if the given two strings are anagrams or not. Return ‘Yes’ if they are anagrams, otherwise, return ‘No’.**

Example  
Input 1: build  
Input 2: dubli  
Output:  
Yes

**Problem Description**

Given a range [A, B], find sum of integers divisible by 7 in this range.

**Problem Constraints**

1 <= A <= B <= 109

**Input Format**

First argument is an integer A.  
Second argument is an integer B.

**Output Format**

Return an integer.

**Example Input**

Input 1:

A = 1

B = 7

Input 2:

A = 99

B = 115

**Example Output**

Output 1:

7

Output 2:

217

**Example Explanation**

Explanation 1:

Integers divisible by 7 in given range are {7}.

Explanation 2:

Integers divisible by 7 in given range are {105, 112}.

**Problem Description**

Find last digit of the number AB.  
A and B are large numbers given as strings.

**Problem Constraints**

1 <= |A| <= 105  
1 <= |B| <= 105

**Input Format**

First argument is a string A.  
First argument is a string B.

**Output Format**

Return an integer.

**Example Input**

Input 1:

A = 2

B = 10

Input 2:

A = 11

B = 11

**Example Output**

Output 1:

4

Output 2:

1

**Example Explanation**

Explanation 1:

210 = 1024, hence last digit is 4.

Explanation 2:

1111 = 285311670611, hence last digit is 1.

**Problem Description**

Find the number of integers in range **[A, B]** with last digit **C**.

**Problem Constraints**

1 <= A <= B <= 109  
0 <= C <= 9

**Input Format**

Given three integers A, B and C.

**Output Format**

Return an integer.

**Example Input**

Input 1:

A = 11, B = 111, C = 1

Input 2:

A = 1, B = 9, C = 0

**Example Output**

Output 1:

11

Output 2:

0

**Example Explanation**

Explanation 1:

The integers are 11, 21, 31, 41, 51, 61, 71, 81, 91, 101, 111

Explanation 2:

There are no integers in the range with last digit 0.

**Problem Description**

Given two integer arrays **A** and **B**, and an integer **C**.  
Find the number of integers in **A which are greater than C** and find the number of integers in **B which are less than C**.  
Return maximum of these two values.

**Problem Constraints**

1 <= |A|, |B| <= 105  
1 <= Ai, Bi, C <=109

**Input Format**

First argument is an integer array A.  
Second argument is an integer array B.  
Third argument is an integer C.

**Output Format**

Return an integer.

**Example Input**

Input 1:

A = [1, 2, 3, 4]

B = [5, 6, 7, 8]

C = 4

Input 2:

A = [1, 10, 100]

B = [9, 9, 9]

C = 50

**Example Output**

Output 1:

0

Output 2:

3

**Example Explanation**

Explanation 1:

There are no integers greater than C in A.

There are no integers less than C in B.

Explanation 2:

Integers greater than C in A are [100].

Integers less than C in A are [9, 9, 9].

**Problem Description**

Given two integers **A** and **B**.  
Find the number of sequences of length **B**, such that every element of this sequence is an positive integer and is **less than of equal to A,** also every previous element in the sequence is less than or equal to half of the next element.

**Problem Constraints**

1 <= A <= 105  
1 <= B <= 20

**Input Format**

Given two integers A and B.

**Output Format**

Return an integer, the number of possible sequences **modulo 109+7**.

**Example Input**

Input 1:

A = 4, B = 2

Input 2:

A = 4, B = 3

**Example Output**

Output 1:

4

Output 2:

1

**Example Explanation**

Explanation 1:

The possible sequences are {1, 2}, {1, 3}, {1, 4}, {2, 4}.

Explanation 2:

The only possible sequence is {1, 2, 4}.

**Problem Description**

Given a string **A**, find the frequency of all the characters in it.

**Problem Constraints**

1 <= |A| <= 105  
Ai = {Lowercase latin alphabets}

**Input Format**

Given a string A.

**Output Format**

Return an integer array of length 26.  
Array should contain frequency of characters in increasing order of characters.

**Example Input**

Input 1:

A = "abcdefghijklmnopqrstuvwxyz"

Input 2:

A = "interviewbit"

**Example Output**

Output 1:

{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}

Output 2:

{0, 1, 0, 0, 2, 0, 0, 0, 3, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 2, 0, 1, 1, 0, 0, 0}

**Example Explanation**

Explanation 1:

Every charcater is present once in the string.

Explanation 2:

'b' is at indices 10.

'e' is at indices 4, 8

'i' is at indices 1, 7, 11

'n' is at indices 2

'r' is at indices 5

't' is at indices 3, 12

'v' is at indices 6

'w' is at indices 9

Rest of the characters are not present in the string

**Problem Description**

Given an array of **A** of length **B×C**.  
Make a **Spiral matrix** (2D array) of B rows and C columns.  
  
Note: See example input for pattern.

**Problem Constraints**

1 <= Ai <=105  
1 <= B×C <= 105

**Input Format**

First argument is an integer array A.  
Second argument is an integer B.  
Third argument is an integer C.

**Output Format**

Return 2D integer array.

**Example Input**

Input 1:

A = [1, 2, 4, 8]

B = 2

C = 2

Input 2:

A = [1, 2, 3, 4, 5, 6, 7, 8, 9]

B = 3

C = 3

**Example Output**

Output 1:

[[1, 2],

[8, 4]]

Output 2:

[[1, 2, 3],

[8, 9, 4],

[7, 6, 5]]

**Problem Description**

Given two integers **A** and **B**, where A is the first element of the sequence then find **Bth** element of the sequence.  
If the **kth** element of the sequence is **X** then **k+1th** element calculated as:

* if X is **even** then next element is **X/2**.
* else next element is **3×X + 1**.

**Problem Constraints**

1 <= A <= 109  
1 <= B <= 105

**Input Format**

Given two integers A and B.

**Output Format**

Return an integer.

**Example Input**

Input 1:

A = 1

B = 3

Input 2:

A = 5

B = 6

**Example Output**

Output 1:

2

Output 2:

1

**Example Explanation**

Explanation 1:

The sequence is as follows 1 -> 4 -> 2

Explanation 2:

The sequence is as follows 5 -> 16 -> 8 -> 4 -> 2 -> 1

**Problem Description**

Given an integer **A**.  
Find the **digital root of A**.  
Digital root is the **repeated sum of digits** of untill there is only one digit left.

**Problem Constraints**

1 <= A <= 109

**Input Format**

Given an integer A.

**Output Format**

Return an integer.

**Example Input**

Input 1:

A = 99

Input 2:

A = 100

**Example Output**

Output 1:

9

Output 2:

1

**Example Explanation**

Explanation 1:

99 -> 9+9 = 18 -> 1+8 = 9

Explanation 2:

100 -> 1+0+0 = 1

**Problem Description**

Given an integer array **A**.  
Create an array **B** such that **Bi** is the **product of all elements of A excluding Ai**.  
Since the products can be too large take **modulo 109 +7**.

**Problem Constraints**

1 <= |A| <= 105  
1 <= Ai <= 109

**Input Format**

Given an integer array A.

**Output Format**

Return an integer array.

**Example Input**

Input 1:

A = [1, 2, 3, 4]

Input 2:

A = [9, 9, 9]

**Example Output**

Output 1:

[24, 12, 8, 6]

Output 2:

[81, 81, 81]

**Example Explanation**

Explanation 1:

[2×3×4, 1×3×4, 1×2×4, 1×2×3]

Explanation 2:

[9×9, 9×9, 9×9]