Accenture Coding Questions | Coding Test for Accenture 2020

September 23, 2020

Accenture Coding Questions 2020 for Freshers

Accenture Coding Test Questions with Answers 2020 are discussed below. A lot of Accenture Coding Question will be of same pattern as mentioned on our Dashboard so it is suggested that you prepare from PrepInsta.

Name of the Section	Coding Assessmneet
Number of Questions	2
Total Time Limit	45 min
Difficulty Level	High

Accenture Dashboard

Accenture Coding Questions

In Accenture there will be 2 coding questions that you have to solve in 45 minutes. In the Accenture Coding Round ,you can write coding using in these **preferred language:-**

- C
- C++
- Java
- Python

The difficulty level of the questions are high. You have to practice alot to get good score in the accenture coding Questions.

Accenture Coding Questions marking Scheme

There will be total of 2 Questions asked in the Accenture Coding Round. For successfully clearing the Coding Round, Students need to have 1 Complete Output and 1 Partial Output.

Accenture Coding Round	No of Questions	Min. Selec
Coding Questions	2	One Com One Part

Rules for Accenture Coding Round Questions Section:

- There are two question for 45 minutes.
- We must start our code from the scratch.
- The coding platform is divided into two, one for writing the code and other for output. We
- The errors are clearly mentioned.
- One Partial and One Complete Output is required for clearing the round.

Accenture Coding Question

Total number of Questions	2 Question
Total Time Duration	45 minutes
Type of Test	Non- Adaptive
Negative Marking	No

Accenture Coding Test Questions and Answers

Question:1

Implement the following Function

def differenceofSum(n. m)

The function accepts two integers n, m as arguments Find the sum of all numbers in range from 1 to m(both inclusive) that are not divisible by n. Return difference between sum of integers not divisible by n with sum of numbers divisible by n.

Assumption:

- n>0 and m>0
- Sum lies between integral range

Example

Input

n:4

m:20

Output

90

Explanation

- Sum of numbers divisible by 4 are 4 + 8 + 12 + 16 + 20 = 60
- Sum of numbers not divisible by 4 are 1 +2 + 3 + 5 + 6 + 7 + 9 + 10 + 11 + 13 + 14 + 15 +
- Difference 150 – 60 = 90

Sample Input

n:3

m:10

Sample Output

19

Python C

```
n = int(input())
m = int(input())
sum1 = 0
sum2 = 0
for i in range(1,m+1):
    if i % n == 0:
        sum1+=i
    else:
        sum2+=i
print(abs(sum2-sum1))
```

```
Input:
3
10
Output:
19
```

Question:2

You are required to implement the following Function

def LargeSmallSum(arr)

The function accepts an integers arr of size 'length' as its arguments you are required to return the sum of second largest largest element from the even positions and second smallest from the odd position of given 'arr'

Assumption:

- All array elements are unique
- Treat the 0th position a seven

NOTE

- Return 0 if array is empty
- Return 0, if array length is 3 or less than 3

Example

Input

arr:3 2 1 7 5 4

Output

7

Explanation

- Second largest among even position elements(1 3 5) is 3
- Second largest among odd position element is 4
- Thus output is 3+4 = 7

Sample Input

arr:1 8 0 2 3 5 6

Sample Output

8

Python C

```
length = int(input())
arr = list(map(int, input().split()))
even_arr = []
odd_arr = []
for i in range(length):
    if i % 2 == 0:
        even_arr.append(arr[i])
    else:
        odd_arr.append(arr[i])
even_arr = sorted(even_arr)
odd_arr = sorted(odd_arr)
print(even_arr[len(even_arr)-2] + odd_arr[len(odd_arr)-2])
```

```
Input:
7
1 8 0 2 3 5 6
Output:
8
```

Question:3

Implement the following Function

def ProductSmallestPair(sum, arr)

The function accepts an integers sum and an integer array arr of size n. Implement the function to find the pair, (arr[j], arr[k]) where j!=k, Such that arr[j] and arr[k] are the least two elements of array (arr[j] + arr[k] <= sum) and return the product of element of this pair

NOTE

- Return -1 if array is empty or if n<2
- Return 0, if no such pairs found
- All computed values lie within integer range

Example

Input

sum:9

Arr:5 2 4 3 9 7 1

Output

2

Explanation

Pair of least two element is (2, 1) 2 + 1 = 3 < 9, Product of (2, 1) 2*1 = 2. Thus, output is 2

Sample Input

sum:4

Arr:9 8 3 -7 3 9

Sample Output

-21

Python C

```
n = int(input())
sum1 = int(input())
arr = list(map(int, input().split()))
if n < 2:
    print('-1')
arr = sorted(arr)
for i in range(n-1):
    if arr[i] + arr[i+1] < sum1:
        print(arr[i] * arr[i+1])
        break
else:
    print('0')
```

```
Input:
6
4
9 8 3 -7 3 9
Output:
-21
```

Question:4

N-base notation is a system for writing numbers which uses only n different symbols, This symbols are the first n symbols from the given notation list(Including the symbol for o) Decimal to n base notation are (0:0, 1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9, 10:A,11:B and so on upto 35:Z)

Implement the following function

Char* DectoNBase(int n, int num):

The function accept positive integer n and num Implement the function to calculate the n-base equivalent of num and return the same as a string

Steps:

1. Divide the decimal number by n,Treat the division as the integer division
2. Write the the remainder (in n-base notation)
3. Divide the quotient again by n, Treat the division as integer division
4. Repeat step 2 and 3 until the quotient is 0
5. The n-base value is the sequence of the remainders from last to first

Assumption:

1 < n < = 36

Example

Input

n: 12

num: 718

Output

4BA

4BA

Explanation

num	Divisor	quotient	remainder
718	12	59	10(A)
59	12	4	11(B)
4	12	0	4(4)

Sample Input

n: 21

num: 5678

Sample Output

CI8

Python

```
n = int(input())
num = int(input())
reminder = []
quotient = num // n
reminder.append(num%n)
while quotient != 0:
    reminder.append(quotient%n)
    quotient = quotient // n
reminder = reminder[::-1]
equivalent = ''
for i in reminder:
    if i > 9:
        a = i - 9
        a = 64 + a
        equivalent+=chr(a)
    else:
        equivalent+=str(i)
print(equivalent)
```

```
Input:
21
5678
Output:
CI8
```

Question:5

Implement the following functions.a

char*MoveHyphen(char str[],int n);

The function accepts a string “str” of length ‘n’, that contains alphabets and hyphens (-). Impler

the string to the front of the given string.

NOTE:- Return null if str is null.

Example :-

- **Input**:

  - str.Move-Hyphens-to-Front
- **Output:**
  - -MoveHyphenstoFront

Explanation:-

The string “Move-Hyphens -to-front” has 3 hyphens (.), which are moved to the front of the string, this output is “— MoveHyphen”

Sample Input

- Str: String-Compare

Sample Output-

- -StringCompare

Python

```
inp = input()
count = 0
final = ""
for i in inp:
    if i == '-':
        count+=1
    else:
        final+=i
print("-"*count,final)
```

```
Output:
move-hyphens-to-front
--- movehyphenstofront
```

Question:6

Problem Statement

A carry is a digit that is transferred to left if sum of digits exceeds 9 while adding two numbers from right-to-left one digit at a time

You are required to implement the following function.

Int NumberOfCarries(int num1 , int num2);

The functions accepts two numbers ‘num1’ and ‘num2’ as its arguments. You are required to c

carries generated while adding digits of two numbers ‘num1’ and ‘ num2’.

Reply here...

Assumption: num1, num2>=0

Example:

- **Input**
  - Num 1: 451

- Num 2: 349
- **Output**
  - 2

Explanation:

Adding 'num 1' and 'num 2' right-to-left results in 2 carries since (1+9) is 10. 1 is carried and (5+4=1) is 10, again 1 is carried. Hence 2 is returned.

Sample Input

Num 1: 23

Num 2: 563

Sample Output

0

C Python

```
#include<stdio.h>

int numberOfCarries(int num1 , int num2)

{

    int carry = 0, sum, p, q, count = 0;

    while((num1!=0)&&(num2!=0))

    {
        p = num1 % 10;
        q = num2 % 10;
        sum = carry + p + q;
        if(sum>9)
        {
            carry = 1;
            count++;
        }
        else
        {
            carry = 0;
        }
        num1 = num1/10;
        num2 = num2/10;
    }
    return count;
}
int main()
{
   int x, y, a;
   scanf("%d",&x);
   scanf("%d",&y);
   a = numberOfCarries(x, y);
   printf("%d",a);
   return 0;
}
```

```
Output:
23
563
0
```

Reply here...

Question:7

Problem Statement

You are given a function,

Void *ReplaceCharacter(Char str[], int n, char ch1, char ch2);

The function accepts a string ' str' of length n and two characters 'ch1' and 'ch2' as its arguments . Implement the function to modify and return the string ' str' in such a way that all occurrences of 'ch1' in original string are replaced by 'ch2' and all occurrences of 'ch2' in original string are replaced by 'ch1'.

Assumption: String Contains only lower-case alphabetical letters.

Note:

- Return null if string is null.
- If both characters are not present in string or both of them are same , then return the string unchanged.

Example:

- **Input:**
  - Str: apples
  - ch1:a
  - ch2:p
- **Output:**
  - Paales

Explanation:

'A' in original string is replaced with 'p' and 'p' in original string is replaced with 'a', thus output is paales.

C Python

```
#include <stdio.h>
#include <string.h>

void *ReplaceCharacter(char str[], int n, char ch1, char ch2)

{

    int i;

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

    {

        if(str[i]==ch1)

        {

            str[i]=ch2;

        }

        else if(str[i]==ch2)

        {

            str[i]=ch1;

        }

    }

    printf("%s",str);

}

int main()

{

    char a[100];

    char b, c;

    int len;

    scanf("%s",a);

    scanf("%s",&b);

    scanf("%s",&c);

    len = strlen(a);

    ReplaceCharacter(a, len, b, c);

    return 0;

}
```

```
Output:
apples
a
p
paales
```

Question:8

Problem Statement

You are required to implement the following function

You are required to implement the following function.

Int OperationChoices(int c, int n, int a , int b)

The function accepts 3 positive integers ‘a’ , ‘b’ and ‘c ‘ as its arguments. Implement the function to return.

- (a+ b) , if c=1
- (a + b) , if c=2
- (a * b) , if c=3
- (a / b) , if c =4

Assumption : All operations will result in integer output.

Example:

- **Input**
  - c :1
  - a:12
  - b:16
- **Output:**
  - Since ‘c’=1 , (12+16) is performed which is equal to 28 , hence 28 is returned.

Sample Input

c : 2

a : 16

b : 20

Sample Output

-4

C Python

```
#include<stdio.h>

int operationChoices(int c, int a , int b)

{

    if(c==1)

    {

        return a + b;

    }

    else if(c==2)

    {

        return a - b;

    }

    else if(c==3)

    {

        return a * b;

    }

    else if(c==4)

    {

        return a / b;

    }

}

int main()

{

    int x, y, z;

    int result;

    scanf("%d",&x);

    scanf("%d",&y);

    scanf("%d",&z);

    result = operationChoices(x, y, z);

    printf("%d",result);

}
```

```
Output:
2
16
20
-4
```

Reply here...

Question:9

Problem Statement

You are given a function,

Int MaxExponents (int a , int b);

You have to find and return the number between 'a' and 'b' (range inclusive on both ends) which has the maximum exponent of 2.

The algorithm to find the number with maximum exponent of 2 between the given range is

1. Loop between 'a' and 'b'. Let the looping variable be 'i'.
2. Find the exponent (power) of 2 for each 'i' and store the number with maximum exponent of 2 so faqrd in a variable , let say 'max'. Set 'max' to 'i' only if 'i' has more exponent of 2 than 'max'.
3. Return 'max'.

Assumption: a <b

Note: If two or more numbers in the range have the same exponents of 2 , return the small number.

Example

- **Input:**
  - **7**
  - **12**
- **Output:**
  - **8**

Explanation:

Exponents of 2 in:

7-0

8-3

9-0

10-1

11-0

12-2

Hence maximum exponent if two is of 8.

Python

```
def countExponents(i):

    count = 0
    while i%2 == 0 and i != 0:

        count+=1

        i = i//2

    return count

def maxExponents(a, b):

    maximum, number = 0, a

    for i in range(a,b):

        temp = countExponents(i)

        if temp>maximum:

            maximum, number = temp, i

    return number

a, b = map(int,input().split())

print(maxExponents(a, b))
```

```
Output:
7 12
8
```

Question : 10

Problem Statement

You are required to implement the following function:

Int Calculate(int m, int n);

The function accepts 2 positive integer ‘m’ and ‘n’ as its arguments.You are required to calculate the sum of numbers divisible both by 3 and 5, between ‘m’ and ‘n’ both inclusive and return the same.

Note

0 < m <= n

Example

Input:

m : 12

n : 50

Output

90

Explanation:

The numbers divisible by both 3 and 5, between 12 and 50 both inclusive are {15, 30, 45} and tl

Sample Input

m : 100

n : 160

Sample Output

405

C C++ Java Python

```
/* Programming Question */

#include <stdio.h>

int Calculate(int, int);

int main()
{
int m, n, result;

// Getting Input

printf("Enter the value of m : ");
scanf("%d",&m);
printf("Enter the value of n : ");
scanf("%d",&n);

result = Calculate(n,m);

// Getting Output

printf("%d",result);

return 0;
}

/* Write your code below . . . */
```

```
int Calculate(int n, int m)
{
// Write your code here

int i, sum = 0;
for(i=m;i<=n;i++)
{
if((i%3==0)&&(i%5==0))
{
sum = sum + i;
}
}

return sum;
}
```

Question 11

Problem Statement

You are required to input the size of the matrix then the elements of matrix, then you have to divide the main matrix in two sub matrices (even and odd) in such a way that element at 0 index will be considered as even and element at 1st index will be considered as odd and so on. then you have sort the even and odd matrices in ascending order then print the sum of second largest number from both the matrices

Example

- **enter the size of array** : 5
- **enter element at 0 index** : 3
- **enter element at 1 index :** 4
- **enter element at 2 index** : 1
- **enter element at 3 index :** 7
- **enter element at 4 index** : 9

Sorted even array : 1 3 9

Sorted odd array : 4 7

10

C **C++** **Java** **Python**

```
#include <stdio.h>

int main()
{
int arr[100];
int length, i, j, oddlen, evenlen, temp, c, d;
int odd[50], even[50];

printf("enter the length of array : ");
scanf("%d",&length);

for(i=0;i<length;i++)
{
printf("Enter element at %d index : ",i);
scanf("%d",&arr[i]);
}

if(length%2==0)
{
oddlen = length/2;
evenlen = length/2;
}
else
{
oddlen = length/2;
evenlen = (length/2) + 1;
}

for(i=0;i<length;i++) // seperation of even and odd array
{
if(i%2==0)
{
even[i/2] = arr[i];
}
else
{
odd[i/2] = arr[i];
}
}

for(i=0; i<evenlen-1; i++) // sorting of even array
{
for(j=i+1; j<evenlen; j++)
{
temp = 0;
temp = 0;
if(even[i]>even[j])
{
temp = even[i];
even[i] = even[j];
even[j] = temp;
}
}
}

for(i=0; i<oddlen-1; i++) // sorting of odd array
{
for(j=i+1; j<oddlen; j++)
{
temp = 0;
if(odd[i]>odd[j])
{
temp = odd[i];
odd[i] = odd[j];
odd[j] = temp;
}
}
}

printf("\nSorted even array : "); // printing even array
for(i=0;i<evenlen;i++)
{
printf("%d ",even[i]);
}

printf("\n");

printf("Sorted odd array : "); // printing odd array
for(i=0;i<oddlen;i++)
{
printf("%d ",odd[i]);
}

printf("\n\n%d",even[evenlen-2] + odd[1]); // printing final result
}
```

Reply here...

Question : 12

Instructions: You are required to write the code. You can click on compile and run anytime to check compilation/execution status. The code should be logically/syntactically correct.

Problem: Write a program in C to display the table of a number and print the sum of all the multiples in it.

Test Cases:

Test Case 1:

Input:

5

Expected Result Value:

5, 10, 15, 20, 25, 30, 35, 40, 45, 50

275

Test Case 2:

Input:

12

Expected Result Value:

12, 24, 36, 48, 60, 72, 84, 96, 108, 120

660

C C++ Java Python

```
 #include <stdio.h>
int main()
{
int n, i, value=0, sum=0;

printf("Enter the number for which you want to know the table: ",n);
scanf("%d",&n);

for(i=1; i<=10; ++i)
{
value = n * i;
printf("table is %d \n",value);
sum=sum+value;
}

printf("sum is %d",sum);

return 0;
}
```

Question : 13

Instructions: You are required to write the code. You can click on compile and run anytime to check compilation/execution status. The code should be logically/syntactically correct.

Question: Write a program in C such that it takes a lower limit and upper limit as inputs and print all the intermediate pallindrome numbers.

Test Cases:

TestCase 1:

Input :

10 , 80

Expected Result:

11 , 22 , 33 , 44 , 55 , 66 , 77.

Test Case 2:

Input:

100,200

Expected Result:

101 , 111 , 121 , 131 , 141 , 151 , 161 , 171 , 181 , 191.

C C++ Java Python

```c
 #include<stdio.h>
int main()
{
int i, n, reverse, d,f,l;
printf("enter the starting \n",f);
scanf("%d",&f);
printf("enter the ending\n",l);
scanf("%d",&l);
for (i = f; i <= l; i++)
{
reverse = 0;
n = num;
while (n != 0)
{
d = n % 10;
reverse = reverse * 10 + d;
n = n / 10;
}
if (i == reverse)
printf("%d ",i);
}
return 0;
}
```

Question : 14

Instructions: You are required to write the code. You can click on compile & run anytime to che the program. The submitted code should be logically/syntactically correct and pass all the tes

Ques: The program is supposed to calculate the distance between three points.

For

x1 = 1 y1 = 1

x2 = 2 y2 = 4

x3 = 3 y3 = 6

Distance is calculated as : **sqrt$(x_2-x_1)^2 + (y_2-y_1)^2$**

C Python

```
#include <stdio.h>
#include <math.h>

int isDistance(float *pt1, float *pt2, float *pt3)
{
float a, b, c;
a = sqrt(((pt2[0]-pt1[0])*(pt2[0]-pt1[0]))+((pt2[1]-pt1[1])*(pt2[1]-pt1[1])));
printf(“%f”,a);
b = sqrt(((pt3[0]-pt2[0])*(pt3[0]-pt2[0]))+((pt3[1]-pt2[1])*(pt3[1]-pt2[1])));
printf(“%f”,b);
c = sqrt(((pt3[0]-pt1[0])*(pt3[0]-pt1[0]))+((pt3[1]-pt1[1])*(pt3[1]-pt1[1])));
printf(“%f”,c);
}

int main()
{
int t;
float p1[2], p2[2], p3[2];
printf("enter x1 and y1 : ");
scanf("%f%f",&p1[0],&p1[1]);
printf("enter x2 and y2 : ");
scanf("%f%f",&p2[0],&p2[1]);
printf("enter x3 and y3 : ");
scanf("%f%f",&p3[0],&p3[1]);
t = isDistance(&p1, &p2, &p3);
printf("%d",t);
return 0;
}
```

Reply here...

Additional Information (FAQ's)

In which all coding languages we can solve the Coding Question asked in Accenture Coding Round?

Students can use any of the following languages to solve the Coding Questions

- C
- C++
- Python
- Java

In which all coding languages we can solve the Codin Accenture Coding Round?

For the complete Online Assessment of the Exam, Accenture uses CoCubes as a platfor

What is the difficulty of the Coding Questions asked in Accenture Coding Test 2020?

The Coding Questions asked in Accenture are of two difficulty type

- 1 Question with Medium to High difficulty
- 1 Question High difficulty

Is PrepInsta enough to prepare for Accenture Coding Round and Questions asked in the exams?

Yes, it is the best resource out there in the internet to prepare for Accenture Coding section paper.

How to Clear Accenture Coding Round?

Prepare for PrepInsta's best Coding Question material, this will help you understand the difficulty of the questions that can be asked in the exam and also Students in PrepInsta's Online class for Accenture will get the opportunity to solve all the previous year questions that were asked in Accenture Coding Round.

Login/Signup to comment

Trishul please add some more questions & solution Prepinsta

3 Log in to Reply

Mohit and is this questions are of high difficulty or medium one ,in respect to accenture exam

0 Log in to Reply

HelpPrepInsta These questions are between easy to moderate

2 Log in to Reply

ishita sir loved it add more questions please!!!!!!!!!!!!!!!!

6 Log in to Reply

shubham prep insta will u be adding more coding questions for accenture ?

1 Log in to Reply

Reply here...

HelpPrepInsta Yes Shubham sure, we'll add some more questions on this page

4
Log in to Reply

Mamta Can I get more questions for practice

4
Log in to Reply

Vaibhav Jain We'll surely update more questions on this page Mamta, please re-visit the page after a few days

2
Log in to Reply

Xtylish Jha Vishesh good question type for practice..

6
Log in to Reply

Support

Contact Us
About Us
Refund Policy
Privacy Policy
Services
Disclaimer
Terms and Conditions

Companies

Accenture
Cognizant
MindTree
VMware
CapGemini
Deloitte
Wipro

Microsoft
TCS
Infosys
Oracle
HCL
TCS Ninja
IBM

All Exams Dashboards

CoCubes Dashboard
eLitmus Dashboard
HirePro Dashboard
MeritTrac Dashboard
Mettl Dashboard
DevSquare Dashboard

Get In Touch

facebook
Twitter
G+
Youtube
Instagram
Linkedin
Telegram

Get In Touch

support@prepinsta.com
+91-8448440710
Text Us on Facebook

PrepInsta.com

No.1 and most visited website for Placements in India.

We help students to prepare for placements with the best study material, online classes, Sectional Statistics for better focus and Success stories & tips by Toppers on PrepInsta.