



HOME

PREVIOUS PAPERS

COGNITIVE ASSESSMENT

COMMON APPLICATION AND MS OFFICE

NETWORKING SECURITY AND

Prime Video

Video

Prime

OffCampus Updates Placement Stats

Accenture Coding Questions and Solution 2024

Accenture Coding Questions With Answers 2024

Accenture Coding Test Questions with Solution 2024 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 Preplinsta.

Preplinsta offer resources specifically tailored to Accenture's coding test patterns. By utilizing these resources, you can gain valuable insights into the types of questions that may be asked during the test.



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
- Dot Net

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. Selection Criteria
Coding Questions	2	One Complete Output One Partial Output

Prime Course Trailer



Related Banners

India's most popular website for Upskilling & Placement Courses

million+ Students got placed

0+ courses under one subscription
Certifications & Projects included

Courses like :-

- ML
- Cloud Computing
- C++
- Java
- Python
- Big Data
- Competitive Coding
- Basic Coding

Join Now

20% off today

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 should write the whole program.
- 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: Rat Count House

(Asked in Accenture OnCampus 10 Aug 2021, Slot 1)

Problem Description :

The function accepts two positive integers 'r' and 'unit' and a positive integer array 'arr' of size 'n' as its argument 'r' represents the number of rats present in an area, 'unit' is the amount of food each rat consumes and each ith element of array 'arr' represents the amount of food present in 'i+1' house number, where $0 \leq i < n$.

Note:

- Return -1 if the array is null
- Return 0 if the total amount of food from all houses is not sufficient for all the rats.
- Computed values lie within the integer range.

Example:

Input:

- r: 7
- unit: 2
- n: 8
- arr: 2 8 3 5 7 4 1 2

Output:

4

Explanation:

Total amount of food required for all rats = $r * \text{unit}$

$$= 7 * 2 = 14.$$

The amount of food in 1st 4 houses = $2+8+3+5 = 18$. Since, amount of food in 1st 4 houses is sufficient for all the rats. Thus, output is 4.

C++

C

Python

Java

Run

```
#include <stdio.h>

int calculate (int r, int unit, int arr[], int n)
{
    if (n == 0)
        return -1;
    int totalFoodRequired = r * unit;
    int foodTillNow = 0;
    int house = 0;
    for (house = 0; house < n; ++house)
    {
        foodTillNow += arr[house];
        if (foodTillNow >= totalFoodRequired)
        {
            break;
        }
    }
    if (totalFoodRequired > foodTillNow)
        return 0;
    return house + 1;
}

int main ()
{
    int r;
    scanf ("%d", &r);
    int unit;
    scanf ("%d", &unit);
    int n;
    scanf ("%d", &n);
    int arr[n];
    for (int i = 0; i < n; ++i)
    {
        scanf ("%d", &arr[i]);
    }
    printf ("%d", calculate (r, unit, arr, n));
    return 0;
}
```

Question 2:

(Asked in Accenture OnCampus 10 Aug 2021, Slot 2)

Problem Description :

The Binary number system only uses two digits, 0 and 1 and number system can be called binary string. You are required to implement the following function:

```
int OperationsBinaryString(char* str);
```

The function accepts a string str as its argument. The string str consists of binary digits separated with an alphabet as follows:

- A denotes AND operation
- B denotes OR operation
- C denotes XOR Operation

You are required to calculate the result of the string str, scanning the string to right taking one operation at a time, and return the same.

Note:

- No order of priorities of operations is required
- Length of str is odd
- If str is NULL or None (in case of Python), return -1

Input:

str: 1C0C1C1A0B1

Output:

1

Explanation:

The alphabets in str when expanded becomes “1 XOR 0 XOR 1 XOR 1 AND 0 OR 1”, result of the expression becomes 1, hence 1 is returned.

Sample Input:

0C1A1B1C1C1B0A0

Output:

0

C++

C

python

Java

Run

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

int OperationsBinaryString (char *str)
{
    if (str == NULL)
        return -1;
    int i = 1;
    int a = *str - '0';
    str++;
    while (*str != '\0')
    {
        char p = *str;
        str++;
        if (p == 'A')
            a &= (*str - '0');
        else if (p == 'B')
            a |= (*str - '0');
        else
            a ^= (*str - '0');
        str++;
    }
    return a;
}

int main ()
{
    char str[100];
    fgets (str, sizeof (str), stdin);
    int len = strlen (str);
    if (str[len - 1] == '\n')
    {
        str[len - 1] = '\0';      // Remove the newline character
        len--;                  // Decrement the length
    }
    int result = OperationsBinaryString (str);
    printf ("%d\n", result);
}
```

```
    printf ("%d\n", result);
    return 0;
}
```

Question 3: Password Checker

(Asked in Accenture OnCampus 10 Aug 2021, Slot 3)

You are given a function.

```
int CheckPassword(char str[], int n);
```

The function accepts string str of size n as an argument. Implement the function which returns 1 if given string str is valid password else 0.

str is a valid password if it satisfies the below conditions.

- At least 4 characters
- At least one numeric digit
- At Least one Capital Letter
- Must not have space or slash (/)
- Starting character must not be a number

Assumption:

Input string will not be empty.

Example:

Input 1:

aA1_67

Input 2:

a987 abC012

Output 1:

1

Output 2:

0

C++ C python Java

Run

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

int CheckPassword (char str[], int n)
{
    // At least 4 characters
    if (n < 4)
        return 0;
    // Starting character must not be a number
    if (str[0] - '0' >= 0 && str[0] - '0' <= 9)
        return 0;
    int a = 0, cap = 0, nu = 0;
    while (a < n)
    {
        // Must not have space or slash (/)
        if (str[a] == ' ' || str[a] == '/')
            return 0;
        // Counting capital letters
        if (str[a] >= 'A' && str[a] <= 'Z')
        {
            cap++;
        }
        // Counting numeric digits
        else if (str[a] - '0' >= 0 && str[a] - '0' <= 9)
        {
            nu++;
        }
        // Incrementing for the while loop
    }
    if (cap > 0 && nu > 0)
        return 1;
    else
        return 0;
}
```

```

        a++;
    }
    // Returns 1 if there are > 0 numeric digits and capital letters
    return cap > 0 && nu > 0;
}

int main ()
{
    char str[100];
    fgets (str, sizeof (str), stdin);
    int len = strlen (str);
    if (str[len - 1] == '\n')
    {
        str[len - 1] = '\0';           // Remove the newline character
        len--;                      // Decrement the length
    }
    int result = CheckPassword (str, len);
    printf ("%d\n", result);
    return 0;
}

```

Question 4:

(Asked in Accenture OnCampus 11 Aug 2021, Slot 1)

You are given a function,

```
int findCount(int arr[], int length, int num, int diff);
```

The function accepts an integer array 'arr', its length and two integer variables 'num' and 'diff'. Implement this function to find and return the number of elements of 'arr' having an absolute difference of less than or equal to 'diff' with 'num'.

Note: In case there is no element in 'arr' whose absolute difference with 'num' is less than or equal to 'diff',
return -1.

Example:

Input:

- arr: 12 3 14 56 77 13
- num: 13
- diff: 2

Output:

3

Explanation:

Elements of 'arr' having absolute difference of less than or equal to 'diff' i.e. 2 with 'num' i.e. 13 are 12, 13 and 14.

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

int findCount (int n, int arr[], int num, int diff)
{
    int count = 0;
    for (int i = 0; i < n; ++i)
    {
```

Run

```

        if (abs (arr[i] - num) <= diff)
        {
            count++;
        }
    }
    return count > 0 ? count : -1;
}

int main ()
{
    int n;
    scanf ("%d", &n);
    int arr[n];
    for (int i = 0; i < n; ++i)
    {
        scanf ("%d", &arr[i]);
    }
    int num;
    scanf ("%d", &num);
    int diff;
    scanf ("%d", &diff);
    printf ("%d\n", findCount (n, arr, num, diff));
    return 0;
}

```

Question 5 :

(Asked in Accenture OnCampus 11 Aug 2021, Slot 2)

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 $1 + 2 + 3 + 5 + 6 + 7 + 9 + 10 + 11 + 13 + 14 + 15 + 17 + 18 + 19 = 150$
- Difference $150 - 60 = 90$

Sample Input

n:3

m:10

Sample Output

A screenshot of a programming environment showing a C code editor. The code implements a function to calculate the difference between the sum of even indices and the sum of odd indices in an array. It includes a main function to read input and print the result. A 'Run' button is visible in the top right corner.

```
#include<stdio.h>
int differenceofSum (int n, int m)
{
    int i, sum1 = 0, sum2 = 0;
    for (i = 1; i <= m; i++)
    {
        if (i % n == 0)
        {
            sum1 = sum1 + i;
        }
        else
        {
            sum2 = sum2 + i;
        }
    }
    if (sum2 > sum1)
        return sum2 - sum1;
    else
        return sum1 - sum2;
}
int main ()
{
    int n, m;
    int result;
    scanf ("%d", &n);
    scanf ("%d", &m);
    result = differenceofSum (n, m);
    printf ("%d", result);
    return 0;
}
```



Question:6

(Asked in Accenture OnCampus 11 Aug 2021, Slot 3)

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 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 as even

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 smallest 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

[C++](#) [C](#) [Java](#) [Python](#)

Run

```
#include <stdio.h>

int LargeSmallSum (int arr[], int length)
{
    if (length <= 3)
    {
        return 0;
    }

    int evenPos[100], oddPos[100];
    int evenCount = 0, oddCount = 0;

    for (int i = 0; i < length; i++)
    {
        if (i % 2 == 0)
        {
            evenPos[evenCount++] = arr[i];
        }
        else
        {
            oddPos[oddCount++] = arr[i];
        }
    }

    // Sort the even position array in descending order
    for (int i = 0; i < evenCount - 1; i++)
    {
        for (int j = 0; j < evenCount - i - 1; j++)
        {
            if (evenPos[j] < evenPos[j + 1])
            {
                int temp = evenPos[j];
                evenPos[j] = evenPos[j + 1];
                evenPos[j + 1] = temp;
            }
        }
    }
}
```

```

        // Sort the odd position array in ascending order
        for (int i = 0; i < oddCount - 1; i++)
        {
            for (int j = 0; j < oddCount - i - 1; j++)
            {
                if (oddPos[j] > oddPos[j + 1])
                {
                    int temp = oddPos[j];
                    oddPos[j] = oddPos[j + 1];
                    oddPos[j + 1] = temp;
                }
            }

            return evenPos[1] + oddPos[1];
        }

        int main ()
{
    int arr[] = { 3, 2, 1, 7, 5, 4 };
    int length = sizeof (arr) / sizeof (arr[0]);

    int result = LargeSmallSum (arr, length);
    printf ("%d\n", result);

    return 0;
}

```

Question:7

(Asked in Accenture OnCampus 12 Aug 2021, Slot 1)

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 ($arr[j] + arr[k] \leq sum$) 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

size of Arr = 7

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

size of Arr = 6

Arr:9 8 3 -7 3 9

Sample Output

-21

C++

C

Java

Python

Run

```
#include<stdio.h>
int productSmallestPair (int *array, int n, int sum)
{
    int answer, temp, i, j, check;
    if (n < 2)
    {
        answer = -1;
    }
    else
    {
        for (i = 0; i < n; i++) //sorting of array
        {
            for (j = i + 1; j < n; j++)
            {
                if (array[i] > array[j])
                {
                    temp = array[i];
                    array[i] = array[j];
                    array[j] = temp;
                }
            }
        }
        check = array[0] + array[1];
        if (check <= sum)
        {
            answer = array[0] * array[1];
        }
        else
        {
            answer = 0;
        }
    }
    return answer;
}
int main ()
{
    int n, sum, result, i;
    scanf ("%d", &sum);
    scanf ("%d", &n);
    int array[n];
    for (i = 0; i < n; i++)
    {
        scanf ("%d", &array[i]);
    }
    result = productSmallestPair (array, n, sum);
    printf ("%d", result);
    return 0;
}
```

Question:8

(Asked in Accenture OnCampus 12 Aug 2021, Slot 2)

N-base notation is a system for writing numbers that uses only n different symbols. These symbols are the first n symbols from the given notation list (Including the symbol for 0). 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 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:

$1 < n \leq 36$

Example

Input

n: 12

num: 718

Output

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

C18

C++

C

Java

Python

Run

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

char *decitoNBase (int n, int num)
{
    char *res = NULL;
    int quotient = num / n;
    int *rem = malloc (sizeof (int));
    int remSize = 1;
    rem[0] = num % n;

    while (quotient != 0)
    {
        remSize++;
        int *temp = realloc (rem, remSize * sizeof (int));
        if (temp == NULL)
        {
            free (rem);
            return NULL;
        }
        rem = temp;
        rem[remSize - 1] = quotient % n;
        quotient = quotient / n;
    }

    int resSize = remSize + 1;
    res = malloc (resSize * sizeof (char));
    if (res == NULL)
    {
        free (rem);
        return NULL;
    }

    int i, j = 0;
    for (i = remSize - 1; i >= 0; i--)
    {
        if (rem[i] > 9)
        {
            res[j] = (char) (rem[i] - 9 + 64);
        }
        else
        {
            res[j] = (char) (rem[i] + '0');
        }
        j++;
    }
    res[j] = '\0';

    free (rem);
    return res;
}

int main ()
{
    int n, num;
    scanf ("%d %d", &n, &num);
    char *result = decitoNBase (n, num);
    printf ("%s\n", result);
    free (result);
    return 0;
}
```

Question:9

(Asked in Accenture Offcampus 1 Aug 2021, Slot 1)

Implement the following functions.

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

The function accepts a string “str” of length ‘n’, that contains alphabets and hyphens (-). Implement the function to move all hyphens(-) in 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

C++

C

Java

Python

Run

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

char * MoveHyphen (char *s, int n)
{
    int count = 0;
    int i = 0;
    while (i < n)
    {
        if (s[i] == '-')
        {
            count++;
            memmove (&s[i], &s[i + 1], n - i);
            n--;
        }
        else
        {
            i++;
        }
    }

    char *res = malloc ((n + count + 1) * sizeof (char));
    if (res == NULL)
    {
        return NULL;
    }

    for (i = 0; i < count; i++)
    {
        res[i] = '-';
    }
    for (i = count; i < count + n; i++)
    {
        res[i] = s[i - count];
    }
}
```

```
        res[count + n] = '\0';

    return res;
}

int main ()
{
    char s[100];
    scanf ("%s", s);
    int n = strlen (s);
    char *result = MoveHyphen (s, n);
    printf ("%s\n", result);
    free (result);
    return 0;
}
```

Question:10

(Asked in Accenture Offcampus 1 Aug 2021, Slot 2)

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 calculate and return the total number of carries generated while adding digits of two numbers 'num1' and ' num2'.

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++

C

Java

Python

Run

```
#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;
    }
    while (num1 != 0)
    {
        p = num1 % 10;
        sum = carry + p;
        if (sum > 9)
        {
            carry = 1;
            count++;
        }
        else
            carry = 0;
        num1 = num1 / 10;
    }
    while (num2 != 0)
    {
        q = num2 % 10;
        sum = carry + q;
        if (sum > 9)
        {
            carry = 1;
            count++;
        }
        else
            carry = 0;
        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;
}
```

Accenture Online Course
PreInsta Prime Video

Crack Accenture with more than 100+ hours of content



Join now

Question:11

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++

C

Java

Python

Run

```
#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;
}
```

Question:12

(Asked in Accenture Offcampus 2 Aug 2021, Slot 1)

Problem Statement

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++

C

Java

Python

Run

```
#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);
    return 0;
}

```

Question:13

(Asked in Accenture Offcampus 2 Aug 2021, Slot 2)

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 far 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 of two is 8.

[C++](#)[C](#)[Java](#)[Python](#)[Run](#)

```
#include <stdio.h>

int count (int n)
{
    int c = 0;
    while (n % 2 == 0 && n != 0)
    {
        c++;
        n = n / 2;
    }
    return c;
}

int maxExponents (int a, int b)
{
    int max = 0, num = 0, ans;
    for (int i = a; i <= b; i++)
    {
        int temp = count (i);
        if (temp > max)
        {
            max = temp;
            num = i;
        }
    }
    return num;
}

int main ()
{
    int a, b;
    scanf ("%d %d", &a, &b);
    printf ("%d", maxExponents (a, b));
    return 0;
}
```

Question : 14

(Asked in Accenture Offcampus 2 Aug 2021, Slot 3)

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

$0 < m \leq n$

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 their sum is 90.

Sample Input

m : 100

n : 160

Sample Output

510



Run

```
#include<stdio.h>
int Calculate (int, int);
int main ()
{
    int m, n, result;
    printf ("Enter the value of m : ");
    scanf ("%d", &m);
    printf ("Enter the value of n : ");
    scanf ("%d", &n);
    result = Calculate (n, m);
    printf ("%d", result);
    return 0;
}
int Calculate (int n, int m)
{
    int i, sum = 0;
    for (i = m; i <= n; i++)
    {
        if ((i % 3 == 0) && (i % 5 == 0))
        {
            sum = sum + i;
        }
    }
    return sum;
}
```

Question 15

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

7

C

C++

Java

Python

Run

```
#include <stdio.h>

int main() {
    int arr[100];
    int length, i, j, oddlen, evenlen, temp;
    int odd[50], even[50];
    printf("Enter the length of the array: ");
    scanf("%d", &length);

    for (i = 0; i < length; i++) {
        printf("Enter element at index %d: ", 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++) {
        if (i % 2 == 0) {
            even[i / 2] = arr[i];
        } else {
            odd[i / 2] = arr[i];
        }
    }

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

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

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

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

```
        }
        printf("\n");

        printf("%d", even[evenlen - 2] + odd[oddlen - 2]);

        return 0;
    }
```

Accenture Online Course PreInsta Prime Video



Join now

Crack Accenture with more than 100+ hours of content

Question : 16

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

Run

```
#include<stdio.h>
int main ()
{
    int n, i, value = 0, sum = 0;
    printf ("Enter the number for which you want to know the table : ");
    scanf ("%d", &n);
    for (i = 1; i <= 10; ++i)
    {
        value = n * i;
        printf ("%d ", value);
        sum = sum + value;
    }
    printf ("\nsum is %d", sum);
    return 0;
}
```

Question : 17

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 palindrome 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

Run

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

Question : 18

Instructions: You are required to write the code. You can click on compile & run anytime to check the compilation/execution status of the program. The submitted code should be logically/syntactically correct and pass all the test cases.

Ques: The program is supposed to calculate the sum of distance between three points from each other.

```
for  
x1 = 1 y1 = 1  
x2 = 2 y2 = 4  
x3 = 3 y3 = 6  
  
Distance is calculated as : sqrt(x2-x1)^2 + (y2-y1)^2
```

C++

C

Java

Python

Run

```
#include<stdio.h>  
#include<int main()  
{  
    float x1,y1,x2,y2,x3,y3;  
    printf("Enter x1,y1 : ");  
    scanf("%f %f",&x1,&y1);  
    printf("Enter x2,y2 : ");  
    scanf("%f %f",&x2,&y2);  
    printf("Enter x3,y3 : ");  
    scanf("%f %f",&x3,&y3);  
    float firstDiff =(float) sqrt (pow (x2 - x1, 2) + pow (y2 - y1, 2));  
    float secondDiff =(float) sqrt (pow (x3 - x2, 2) + pow (y3 - y2, 2));  
    float thirdDiff =(float) sqrt (pow (x3 - x1, 2) + pow (y3 - y1, 2));  
    printf("%f", (firstDiff + secondDiff + thirdDiff));  
    return 0;  
}
```

Question : 19

Find the maximum value and its index in the array

Problem Statement :

You are given a function, void MaxInArray(int arr[], int length); The function accepts an integer array 'arr' of size 'length' as its argument. Implement the function to find the maximum element of the array and print the maximum element and its index to the standard output

(STDOUT). The maximum element and its index should be printed in separate lines.

Note:

- Array index starts with 0
- Maximum element and its index should be separated by a line in the output
- Assume there is only 1 maximum element in the array
- Print exactly what is asked, do not print any additional greeting messages

Example:

Input:

23 45 82 27 66 12 78 13 71 86

Output:

86

9

Explanation:

86 is the maximum element of the array at index 9.

```

#include <stdio.h>
#include <limits.h>

void MaxInArray (int arr[], int length)
{
    int max = INT_MIN, index = -1;
    for (int i = 0; i < length; i++)
    {
        if (arr[i] > max)
        {
            max = arr[i];
            index = i;
        }
    }
    printf ("%d\n%d\n", max, index);
}

int main ()
{
    int n;
    scanf ("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
        scanf ("%d", &arr[i]);
    MaxInArray (arr, n);
    return 0;
}

```

Run

Question : 20

Autobiographical Number

Problem Statement :

An Autobiographical Number is a number N such that the first digit of N represents the count of how many zeroes are there in N, the second digit represents the count of how many ones are there in N and so on.

You are given a function, **def FindAutoCount(n):**

The function accepts string “n” which is a number and checks whether the number is an autobiographical number or not. If it is, an integer is returned, i.e. the count of distinct numbers in ‘n’. If not, it returns 0.

Assumption:

- The input string will not be longer than 10 characters.
- Input string will consist of numeric characters.

Note:

If string is None return 0.

Example:

Input:

n: “1210”

Output:

3

Explanation:

0th position in the input contains the number of 0 present in input, i.e. 1, in 1st position the count of number of 1s in input i.e. 2, in 2nd position the count of 2s in input i.e. 1, and in 3rd position the count of 3s i.e. 0, so the number is an autobiographical number.

Now unique numbers in the input are 0, 1, 2, so the count of unique numbers is 3. So 3 is returned.

C++ C Java

Run

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

int FindAutoCount (char n[])
{
    int sum = 0;
    int length = strlen (n);
    int uniqueDigits[10] = { 0 };

    for (int i = 0; i < length; i++)
    {
        sum += (n[i] - '0');
        uniqueDigits[n[i] - '0'] = 1;
    }

    if (sum != length)
        return 0;

    int count = 0;
    for (int i = 0; i < 10; i++)
    {
        if (uniqueDigits[i] == 1)
            count++;
    }

    return count;
}

int main ()
{
    char n[100];
    scanf ("%s", n);
    printf ("%d", FindAutoCount (n));
    return 0;
}
```

Accenture Online Course
PreInsta Prime Video



Join now

FAQs on Accenture Coding Questions and Solution 2024

Question 1: 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
- Dot Net

Question 2: In which all coding languages we can solve the Coding Question asked in Accenture Coding Round?

For the complete Online Assessment of the Exam, Accenture uses CoCubes as a platform to conduct the exam.

Question 3: What is the difficulty of the Coding Questions asked in Accenture Coding Test 2024?

The Coding Questions asked in Accenture are of two difficulty type

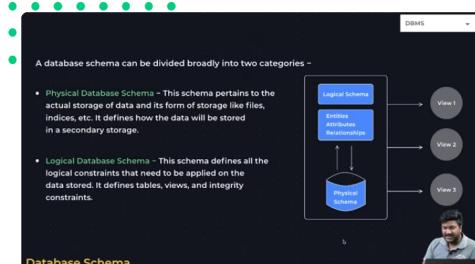
- 1 Question with Medium to High difficulty
- 1 Question High difficulty

Question 3: How to Clear Accenture Coding Round?

Prepare for PreInsta'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 PreInsta's Online class for Accenture will get the opportunity to solve all the previous year questions that were asked in Accenture Coding Round.

Get over 200+ course One Subscription

Courses like AI/ML, Cloud Computing, Ethical Hacking, C, C++, Java, Python, DSA (All Languages), Competitive Coding (All Languages), TCS, Infosys, Wipro, Amazon, DBMS, SQL and others



Get Prime

Course List

Checkout list of all the video courses in PreInsta Prime Subscription

[Checkout](#)

[Login/Signup](#) to comment

A

[Amruta](#) Question no 10's answer is wrong it should be as below

```
def NumberOfCarries(n1,n2):  
    count=0  
    carry = 0  
    l = len(str(n1))  
    if len(str(n1)) >= len(str(n2)):  
        l = len(n2)  
        for i in range(l+1):  
            if ((n1%10 + n2%10 + carry) /10) > 0:  
                count += 1  
            carry = int((n1%10 + n2%10 + carry) /10)  
            n1 = int(n1/10)  
            n2 = int(n2/10)  
    return count  
n1=int(input())  
n2=int(input())  
print(NumberOfCarries(n1,n2))
```

 0 [Log in to Reply](#)



[suraj](#) def a(n):
l=[]
sum=0
for i in n:
l.append(ord(i)-48)
for i in range(len(l)):
if l[i]>0:
sum+=i
return(sum)
n=input()
print(a(n))

 0 [Log in to Reply](#)



[Sakshi](#) My Q-19 python solution (is it correct?)

```
arr = list(map(int,input().split()))  
arr.sort()  
print(arr)  
print(arr[-1])  
print(len(arr)-1)
```

 0 [Log in to Reply](#)



[suraj](#) no

u just change the previous positions of arr

 Log in to Reply



arnab.nandagoswami Thank you so much PreInsta. I got placed in Accenture because of you guys!

 9 Log in to Reply



Vaibhav Jain Yes arnab, but we would recommend that you must visit those company specific coding pages, that would be more relevant

 3 Log in to Reply



HelpPreInsta Hey Arnab!

Congratulations on getting placed, we wish you all the best for your future.

Also, guys if you are looking for Accenture Preparation, check out this link.

[Click here](#)

 0 Log in to Reply



arghyadeep In question number 19 it is asked to find maximum element of an array and its index.

the code given here is wrong.

The code should be-

#include

using namespace std;

void MaxInArray (int arr[],int length)

{

int max=arr[0];

int maxIdx=0;

for (int i = 0; i < max){

maxIdx = i;

max = arr[i];

}

}

cout << max << endl <> n; int arr[n]; for (int i = 0; i < arr.length(); MaxInArray (arr, n); return 0;

}

 7 Log in to Reply



HelpPreInsta Thank you for the submission.

 0 Log in to Reply



Himanshu Sir could you please let me know from where can I get off campus updates so that I don't miss out on any single opportunity?

 4 Log in to Reply



Vaibhav Jain There is no surety of that, but yes there are high chances that you may find a few questions similar to the questions given above

 0 Log in to Reply



HelpPreInsta Hey Himanshu, don't worry we are right here with all the off campus updates.

[Click here](#)

 0 Log in to Reply



jessica Sir there has been so many questions of the same pattern you taught, Preplinsta has been so helpful in the whole learning process for me.

0 [Log in to Reply](#)



HelpPreplinsta Hey Jessica, thank you so much for your valuable feedback.

Also, guys if you looking for placement preparation course which covers whole aptitude, Python, Java DSA, Competitive Coding etc. and past year's Accenture questions along with Interview Preparation material, [Click here](#)

0 [Log in to Reply](#)



bhavesh Team, I am in a need of support from your mentoring team, how can I connect with you guys for instant response?

1 [Log in to Reply](#)



HelpPreplinsta Hey Bhavesh, Our mentoring team is available on 8448440710 via call and on 7044817397 via whatsapp.

Do contact us, wed love to help you out 😊

0 [Log in to Reply](#)



MINAL when can i apply for accenture off-campus drive for 2022 batch?

1 [Log in to Reply](#)



HelpPreplinsta Hey Minal, stay updated on our Instagram and Telegram groups, you'll get all the updates from there regarding Accenture hiring and whole recruitment process as well.

[Click here](#)

0 [Log in to Reply](#)



Raghvendra I have purchased Accenture Prime Mock Material but i am unable to view after buying

0 [Log in to Reply](#)



HelpPreplinsta Hey Raghvendra, if you are facing any issues regarding access, you can call us at 8448440710 or whatsapp message us on 7044817397.

Our team would love to help you out!

0 [Log in to Reply](#)



guttikonda I brought Accenture Prime mock material yesterday how can i access it

where are the mock test i could not find please help

0 [Log in to Reply](#)



HelpPreplinsta Hey lalitha, we are sorry for the inconveniences, you can access it from here.

[Click here](#)

0 [Log in to Reply](#)

[« Previous](#) 1 2



Preplinsta

[Preplinsta.com](#)

Support

[Contact Us](#)

[About Us](#)

[Refund](#)

Companies

[Accenture](#) [Microsoft](#)

[Cognizant](#) [TCS](#)

[MindTree](#) [Infosys](#)

All Exams

[Dashboards](#)

[CoCubes](#)

[Dashboard](#)

Get In

[Touch](#)

[Instagram](#)

[Linkedin](#)

Get In Touch

[support@prepinsta.com](#)

[insta.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 Preplinsta.

Policy

VMware

Oracle

eLitmus Dashboard

Youtube

+91-

Privacy

CapGemini HCL

HirePro Dashboard

Telegram

8448440710

Policy

Deloitte

TCS Ninja

MeritTrac

facebook

Text us on

Services

Wipro

IBM

Dashboard

Twitter

Whatsapp/Insta

Disclaimer

Mettl Dashboard

Terms and

DevSquare

Conditions

Dashboard

© 2022 Prep Insta

[Privacy Policy](#) | Copyright © 2022 Prep Insta

