10 seconds.
Siddarth

C
3
10/02/25

Competitive Coding. | Python, C, C++, Java |

concept based - story based - scenario based

1. Write a function to accept two positive integers n1 & n2 where n1 and n2 should be validated, if ~~is~~ not greater than zero, the function should return zero. n1 should be lesser than n2. Validate that if n1 is not lesser than n2, the function should return -1. Generate natural numbers ranging from n1 to n2 and take the sum of even numbers in the generated natural numbers and take the sum of odd numbers in the generated natural numbers. Finally return the difference between sum of even numbers and sum of odd numbers.

$2+4+6+8+10$
$3+5+7+9$

```
= #include <stdio.h>
int functioncomp(int n1, int n2){
        if(n1<=0 || n2<=0)
                return 0;
        if(n1>=n2)
                return -1;
        ~~else if(n1<n2){~~
        int x;
        for(int i=n1;i<=n2;i++)
                if(i%2 == 0)
                        x += i;
                else
                        x -= i;
        return x;
}
```

[illegible]

Python — ML, AI, DSA
c/c++ — faster & efficient running.
Java — applications/multi-threading.

RAHUL

longest Substring paliadrom

PAGE NO 3.
DATE 10/02/25

2. ceaser cipher for 'SBIVN' using the key -1.

```
=  void
   ~~char[]~~ decrypt (char name[]){
        int key= -1;
        for(int i=0; name[i] != '\0' ; i++){
                name[i] = name[i]-1;
        }       printf("%c", name[i]);
        ~~return name;~~ }
   }
```

3. Sum of digits in string using ASCII code.

```
= #include<stdio.h>
  int main(){
       char number[] = "876548";
       int x=0;
       for(int i=0; number[i]!='\0'; i++)
               x = x+number[i]-48;
        printf("%d", x);
  }
```

~~10 seconds~~
Siddarth

PAGE NO. 3
DATE 12/09/25

Competitive coding.

1. A function accepts 3 parameters 'r', unit, houses. 'r' is a +ve integer, unit is a +ve, houses is a +ve integer array. 'r' represents number of rats present in the city. 'unit' represents the amount of food consumed by each rat ~~Hous~~ ith element in the houses represents the amount of food present in each house. Considering all the rats in the city, start consuming the food right from the first house. return number of houses sufficient for rats in terms of food.

= r = 7, unit = 2, houses = [5, 3, 4, 6, 7, 1]

```
~~int countnhouses(int r, int unit, int houses[]){
    if (r<0 || unit<0 || houses.){
            cout<<"Not a valid input";
    }       return;
    }
```

```
    ~~int n;~~
    ~~for(int i=0;~~

#include<iostream>
using namespace std;
int countnhouses(int r, int unit, int houses[], int n){
    if (r<0 || unit<0 || n<0){
            cout<<"Not a valid input";
            return 0;
    }
    int count=0;
    int total= r* unit;
    for(int i=0;i<n;i++){
        if (total >0){
            total -= houses[i];
            count++;
        }
        }
        return count;
}
int main(){
    int a[]= {5,3,4,6,7,1};
    int n= countnhouses(7,2,a,6);
    count<<n;
}
```

2. Take an array and move all the zero's to the end of the array without changing the order of the initial array.

```
#include<iostream>
using namespace std;
int main(){
    int arr[]= {5,6,0,7,0,9,0,4};
    int size= sizeof(arr)/sizeof(arr[0]);
    for(int i=0; i<size; i++)
        cout<< arr[i] <<"\t";
    cout<<endl;
    int count=0, temp=0;
    for(int i=0; i<size; i++)
        for(int j=0; j<size-1 ;j++)
            if (arr[j]==0){
                temp = arr[j];
                arr[j]= arr[j+1];
                arr[j+1]= temp;
    for (int i=0; i<size ; i++)
        cout<< arr[i] <<'\t";
    cout<<endl;
    return 0;
}
```

10 Seconds
Siddarth.

Competitive Coding.

1. Write a function to return 1, 0 on anagrams. Check whether the two words are anagrams.

=

//Anagram is a state of strings where character count of word1 is equal to the character count of word2.

Eg: Character count of a in word1 is 4 & in word2 is 4

```
#include<iostream>
using namespace std;
int main(){
    String s1 = "abbaa";
    String s2 = "aaaba";
    if (s1.size() != s2.size()){
            cout<< "0", return 0;
    }
     int i=0, n, j=0, x=s1.size(), tch, count=0;
     for(i=0; i<x; i++){
          tch = s1[i];
          count=0;
          for(int j=0; j<x; j++)
               if (tch==s2[j]) count++;
          for(int j=0; j<x; j++)
               if (tch==s1[j]) count--;
          if (count!= 0){
               cout<< "0";
               return0;
          }
     }
     cout<<"1";
     return 1;
}
```

13/02/25

2. Display the count of occurance of each character in a nam

```
→ #include<iostream>
using namespace std;
int main(){
      string s1 = "GaureshbPai";
      for(int i=0;i<s1.size();i++){
            char ch=s1[i];
            int count=0;
            for(int j=0;j<s1.size();j++){
                  if(ch==s1[j]) count++;
            cout<<s1[i]<<" : "<<count<<endl;
      }
      return 1;
}
```

3. Sum of the digits present in the integer.

```
= #include <iostream>
using namespace std;
int main(){
      int n=111, sum=0;
while(n!=0)~~for(int i=0;n!=0;i++)~~{
            sum += (n%10);
            n=n/10;
      }
      cout<<sum;
      return 0;
}
```

③
371

10 Seconds
Siddarth

14/02/25

Competitive Coding

1. Write the function to count the number of carry digits.
=
```
#include<iostream>
using namespace std;
int main(){
    int a= 9999, b= 1, count =0, carry=0;
    while(a>0 || b>0){
        if(a%10+b%10+carry)>9){
            count ++;
            carry=1;
        }else{
            carry=0;
        }
        a/=10;
        b/=10;
    }
    cout<<count;
}
```

2. Write the function to print the rotated array of a given array by n times.
=
```
#include<iostream>
using namespace std;
void printRotatedArray(int a[], int n, int size1){
    int b[size1], x=0;
    for (int i= (n%size1); i<size1; i++){
        b[x]= a[i];
        x= (x+1)%size1;
    }
    for(int i=0; i<(n%size1); i++){
        b[x]= a[i];
        x=(x+1)% size1;
    }
```

```
        for(int i=0;i<size1;i++)
            cout<<b[i]<<"\t";
}
int main(){
        int a[]={50,10,20,30,40};
        int n=11;
        int size1= size(a);
        printRotatedArray(a,n,size1);
}
```

3. Find the last element of the triangle of size n where the first element is 1 and the next element is the sum of the previous two elements from the column of the previous column except the first column where the elements are the last element from the previous row.

=
```
#include <iostream>
using namespace std;
int main(){
        int n=5,a[n][n];
        a[0][0]=1;
        for(int i=1;i<n;i++)
            for(int j=0;j<i+1;j++)
                if(j==0) a[i][j]=a[i-1][i-1];
                else a[i][j]=a[i][j-1]+a[i-1][j-1];
        cout<<a[n-1][n-1];
}
```

4. SQL queries.

=
```
create table Students(
        id int primary key AUTOINCREMENT,
        student_name varchar(20),
        marks INT
)
```

```
insert into students values (1,'Sachin',89),(2,'Rahul',75),
                                  (3,'Shrrithi',85);

-- display the last inserted id without using aggregate functio
select id from students order by id desc limit 1;

-- display the student name with the marks greater
-- than avg
select student_name from students where marks >
        (select avg(marks) from students);

-- display the students whose admission is done
create table admissions (
        id int primary key,
        student_id int,
        ad_no int,
        FOREIGN KEY(student_id) REFERENCES students
)

insert into admissions VALUES (1,3,123),(2,2,234);

select student_name from students, admissions where
        student_id = students.id;
```