TCS NQT PREVIOS YEAR CODING QUESTIONS

1. Consider the below series:

1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17, ...

This series is a mixture of 2 series - all the odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order.

Write a program to find the Nth term in this series.

The value N is a Positive integer that should be read from STDIN. The Nth term that is calculated by the program should be written to STDOUT. Other than the value of Nth term, no other characters/strings or message should be written to STDOUT.

For example, when N = 14, the 14th term in the series is 17. So only the value 17 should be printed to STDOUT.

```
#include<iostream>

using namespace std;

#define MAX 1000


void fibonacci(int n)

{

        int i, t1 = 0, t2 = 1, nextTerm;

        for (i = 1; i<=n; i++)

        {

                nextTerm = t1 + t2;

                t1 = t2;

                t2 = nextTerm;

        }

        cout << t1;

}

void prime(int n)

{

        int i, j, flag, count =0;
```

```
    for (i=2; i<=MAX; i++)

    {

        flag = 0;

        for (j=2; j<i; j++)

        {

            if(i%j == 0)

            {

                flag = 1;

                break;

            }

        }

        if (flag == 0)

        if(++count == n)

        {

            cout << i;

            break;

        }

    }

}

int main()

{

    int n;

    cin >> n;

    if(n%2 == 1)

        fibonacci (n/2 + 1);

    else

        prime(n/2);

return 0;

}
```

2. Given a series whose even term creates a separate geometric series and odd term creates another geometric series.

Write a program to generate such series.For example,1, 1, 2, 2, 4, 4, 8, 8, 16, 16,......

```
#include<iostream>

using namespace std;


int main()

{

    int n, i, r1, r2;

    cout << "nEnter the total number of terms : ";

    cin >> n;

    cout << "nEnter the common ratio for GP - 1 : ";

    cin >> r1;

    cout << "nEnter the common ratio for GP - 2 : ";

    cin >> r2;

    cout << "nThe series isn";

    int a = 1, b = 1;

    if(n % 2 == 0)

    {

        for(i = 0; i < n/2; i++)

        {

            cout << a << " ";

            a = a * r1;

        cout << b << " ";

            b = b * r2;

        }

    }

    else
    {
        for(i = 0; i < n/2; i++)
        {
            cout << a << " ";
            a = a * r1;
            cout << b << " ";
            b = b * r2;;
        }
    cout << a << " ";
    }
    cout << endl;
}
```

3. Finding area of a circle

Program to find the area of a circle is discussed here. Area of the circle can be found using the formula, $A = \pi r^2$ where r is the radius of the circle. When the radius of the circle is known, the area of the circle can be calculated using the formula mentioned.

```
#include <iostream>
#include <math.h>
using namespace std;

float area_of_a_circle(float radius)
{
return M_PI * radius * radius;
}

int main()
{
```

```
float area, radius;
cout << “\nEnter the radius of the circle : “;
cin >> radius;
area = area_of_a_circle(radius);
cout << “\nArea of the circle : ” << area <<endl;
return 0;
}
```

4.Checking if a given year is leap year or not

```
#include <iostream>
using namespace std;
int main()
{
int year;
cout << “Enter a year: “;
cin >> year;
cout << endl;
if(year%4 == 0)
{
if( year%100 == 0)
{
if ( year%400 == 0)
cout << year << ” is a leap year” ;
else
cout << year << ” is not a leap year”;
}
else
cout << year << ” is a leap year”;
}
else
cout << year << ” is not a leap year”;
cout << endl;
return 0;
}
```

5. GCD of two numbers | Program to find the GCD or HCF of two numbers

```
#include<iostream>
using namespace std;
int main()
{
int a,b,gcd;
cout <<“\nEnter two numbers : “;
cin >> a >> b;
int i;
for(i = 1; i <= a && i <= b; i++)
{
if((a % i == 0) && (b % i == 0))
{
gcd = i;
}
}
cout << “\nGCD of “<< a << ” and ” << b << ” is ” << gcd;
cout << endl;
return 0;
}
```

6. Check whether a given number is a prime or not

```
#include<stdio.h>
int main()
{
int n,i;
printf(“\nEnter the number : “);
scanf(“%d”,&n);
for(i = 2; i <= n/2; i++)
{
if(n % i ==0)
{
break;
}
}
if(i > n/2)
printf(“\n%d is a Prime Number\n”,n);
else
printf(“\n%d is not a Prime Number\n”, n);
return 0;
}
```

7.Find prime numbers in a given range

```
#include <iostream>
```

```
using namespace std;

int main()

{

int a, b, i, flag;

cout << “\nEnter start value : “;

cin >> a;

cout << “\nEnter end value : ” ;

cin >> b;

cout << “\nPrime Numbers between ” << a << ” and ” << b <<” : “;

while (a < b)

{

flag = 0;

for(i = 2; i <= a/2; ++i)

{

if(a % i == 0)

{

flag = 1;

break;

}

}

if (flag == 0)

cout << a << ” “;

++a;

}

cout << endl;

return 0;

}
```

8. Program to check if a given number is a strong number or not

```
#include<iostream>
using namespace std;

int main()
{
int n,i;
int fact,rem;
cout << "\nEnter a number : ";
cin >> n;
cout << endl;
int sum = 0;
int temp = n;
while(n)
{
i = 1,fact = 1;
rem = n % 10;

while(i <= rem)
{
fact = fact * i;
i++;
}
sum = sum + fact;
n = n / 10;
}
if(sum == temp)
    cout << temp << " is a strong number\n";
else
    cout << temp << " is not a strong number\n";

return 0;
}
```

9. Check whether a number is PALINDROME or Not

```
#include<iostream>
using namespace std;

int is_Palindrome(int );
int n;

int main()
{
int palindrome;
cout << “\n\nEnter a number : “;
cin >> n;
cout << endl;
palindrome = is_Palindrome(n);
if(palindrome == 1)
cout << n << ” is a palindrome\n”;
else
cout << n << ” is not a palindrome\n”;
return 0;
}

int is_Palindrome(int aj)
{
static int sum = 0;
if(aj != 0)
{
sum = sum *10 + aj⍰;
is_Palindrome(aj/10); // recursive call
}
else if(sum == n)
return 1;
else
return 0;
}
```

10. Check whether the number is armstrong or not

```
#include<iostream>
#include using namespace std;

int main()
{
int number, temp, remainder, result = 0, n = 0 ;

cout << “Enter an integer : “;
```

```
cin >> number;

temp = number;

// Finding the number of digits

while (temp != 0)
{
temp /= 10;
++n;
}

temp = number;

// Checking if the number is armstrong

while (temp != 0)
{
remainder = temp⍰;
result += pow(remainder, n);
temp /= 10;
}

if(result == number)
cout << number << ” is an Armstrong number\n”;
else
cout << number << ” is not an Armstrong number\n”;
```

```
return 0;
}
```

11. Print the Armstrong numbers between two intervals

```
// C++ program to print the Armstrong numbers between the two intervals

#include <iostream>
#include <math.h>
using namespace std;

int main()
{
int start, end, i, temp1, temp2, remainder, n = 0, result = 0;

cout << “Enter start value and end value : “;
cin >> start >> end;
cout << “\nArmstrong numbers between ” << start << ” and ” << end << ” are : “;

for(i = start + 1; i < end; ++i)
{
temp2 = i;
temp1 = i;

while (temp1 != 0)
{
temp1 /= 10;
++n;
}

while (temp2 != 0)
{
remainder = temp2 % 10;
result += pow(remainder, n);
temp2 /= 10;
}

if (result == i) {
cout << i << ” “;
}

n = 0;
result = 0;

}
cout << endl;
return 0;
}
```

12. Fibonacci series upto n value

```
// C++ program to generate fibonacci series upto n value

#include<iostream>
using namespace std;
int main()
```

```
{
int sum = 0, n;
int a = 0;
int b = 1;
cout << "Enter the nth value: ";
cin >> n;
cout << "Fibonacci series: ";
while(sum <= n)
{
cout << sum << " ";
a = b;  // swap elements
b = sum;
sum = a + b;  // next term is the sum of the last two terms
}
return 0;
}
```

13. Convert the given Binary Number into Decimal

```
// C++ program to convert a binary number into decimal number

#include
#include using namespace std;

int binary_to_decimal(long int n)
{
int decimal = 0, i = 0, remainder;
while (n!=0)
{
remainder = n⍰;
n /= 10;
decimal += remainder*pow(2,i);
++i;
}
return decimal;
}

int main()
{
long int n;
cout << “Enter a binary number: “;
cin >> n;
cout << “\nDecimal number : ” << binary_to_decimal(n) << endl;
return 0;
}
```

Similarly Try the below conversion

14. Decimal to binary conversion

15. Decimal to octal conversion

16. Octal to decimal conversion

17. Binary to octal conversion

18. Octal to binary conversion

19. Find prime numbers in a given range

C++ program to find prime numbers in a given range

```
#include <iostream>
using namespace std;
int main()
{
int a, b, i, flag;
cout << “\nEnter start value : “;
cin >> a;
cout << “\nEnter end value : ” ;
cin >> b;
cout << “\nPrime Numbers between ” << a << ” and ” << b <<” : “;
while (a < b)
{
flag = 0;
for(i = 2; i <= a/2; ++i)
{
if(a % i == 0)
{
flag = 1;
break;
}
}
if (flag == 0)
cout << a << ” “;
++a;
}
cout << endl;
return 0;

}
```

20. Program to Reverse a Number

```
// C++ program to reverse a number

#include <iostream>

using namespace std;

int main()

{

int n, rev = 0, rem;

cout << “\nEnter a number : “;

cin >> n;

cout << “\nReversed Number : “;

while(n != 0)

{

rem = n⍰;

rev = rev*10 + rem;

n /= 10;

}

cout << rev << endl;

return 0;

}
```

21. Program to reverse a string

```
//program to reverse a string in C++

#include <iostream>
using namespace std;
int main()
{
char str[1000], rev[1000];
int i, j, count = 0;
cin >> str;

//finding the length of the string by counting
while (str[count] != '')
{
count++;
}
j = count - 1;

//reversing the string by swapping
for (i = 0; i < count; i++)
{
rev[i] = str[j];
j--;
}
cout << rev;
}
```

22. Pyramid pattern using stars or Pyramid star patterns

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

```
// C++ program to print half pyramid pattern using stars
#include <iostream>
using namespace std;

int main()
{
int i, j,n;
cin >> n;
for(i = 0; i < n; i++)
{
for(j = 0; j <= i; j++)
{
cout << “*”;
}
cout << endl;
}
return 0;
}
```

23.Diamond pattern printing using stars

```
     *
    * *
   * * *
  * * * *
 * * * * *
  * * * *
   * * *
    * *
     *
```

```
/* C++ program – solid diamond pattern printing using stars */
#include <iostream>
using namespace std;

int main()
{
        int n, c, k, space = 1;

        cout << “\nEnter the number of rows : “;
        cin >> n;

        space = n – 1;

        for (k = 1; k <= n; k++)
        {
                for (c = 1; c <= space; c++)
                        cout << ” “;
```

```
    space--;
        for (c = 1; c <= 2*k-1; c++)
            cout << “*”;
        cout << endl;
}
    space = 1;
    for (k = 1; k <= n – 1; k++)
    {
        for (c = 1; c <= space; c++)
            cout << ” “;
        space++;
        for (c = 1 ; c <= 2*(n-k)-1; c++)
   cout << “*”;
        cout << endl;
    }
    return 0;
}
```

24. Program to find the second smallest element in an array

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
int n,i;
cout << "Enter the number of elements : ";
cin >> n;
int arr[n];
cout << "\nInput the array elements : ";
for(i = 0; i < n; i++)
{
cin >> arr[i];
}
sort(arr, arr+n);
cout << "\nThe second largest element is " << arr[1];
cout << endl;
return 0;
}
```

25. Program to remove duplicate elements in an sorted array.

```
/* C++ program to remove duplicate elements in an array */
#include<iostream>
using namespace std;

int remove_duplicate_elements(int arr[], int n)
{

if (n==0 || n==1)
return n;

int temp[n];

int j = 0;
int i;
```

```
for (i=0; i<n-1; i++)
if (arr[i] != arr[i+1])
temp[j++] = arr[i];
temp[j++] = arr[n-1];

for (i=0; i<j; i++)
arr[i] = temp[i];

return j;
}


int main()
{
int n;
cin >> n;
int arr[n];
int i;
for(i = 0; i < n; i++)
{
cin >> arr[i];
}

n = remove_duplicate_elements(arr, n);


for (i=0; i<n; i++)
cout << arr[i] << ” “;
```

```
return 0;

}
```