

## Mindtree Coding Section

**Coding section** consists of 2 questions from coding , wherein the student is required to code in any language of their choice

1) Pattern

for n=5

1

3\*2

4\*5\*6

10\*9\*8\*7

11\*12\*13\*14\*15

1

2\*3

4\*5\*6

2) for n=4,s=3

3

44

555

6666

6666

555

44

3

3) Print all the prime numbers which are below the given number separated by comma.

4) Remove all the vowels from a given string using pointers concept

5) Arrays and pointers:

Merge sort using array pointer.

6) A pattern.

1

22

333

4444

55555

4444

333

22

1

7. Write a function to return a sorted array after merging two unsorted arrays, the parameters will be two integer pointers for referencing arrays and two int variable, the length of arrays (Hint: use malloc() to allocate memory for 3rd array):

8. calculate GCD of two integers a and b which is a direct question and second was merge sort in Array

9. Implement Round-Robin scheduling.

10. Check two binary strings are gray code or not.

**11. GCD of elements in an array.**

12. Arranging K number of elements of an array in ascending order, remaining in descending order.

13. print pattern. void Pattern (int a, int b).

```
2
33
444
5555
5555
444
33
2
```

where a =2 & b=4

14. factorial of number of elements minus 1 programme (n!-1)

15. string manipulation, pointers, factorial

16. code includes If, else, loops, string, functions

17. Given three die, print the possible count of a number.

Example , input : 2 output : 0 because there's no combination that sums up 2 same is number 1. For 3 there's only combination (1,1,1)

18. Display all prime number before 200.

19. string manipulation.

20. code of matrix multiplication

**21. you have three dice each side is numbered as 1,2,3,4,5,6 .you have to take a number from user check that number as sum of any side number of these three dice.like we take number as 5 then the possibility are (1,1,3) (1,2,2) (1,3,1) (2,1,2) (2,2,1) (3,1,1) and is 6 possibility. So write to a program for that. You can write program in c.**

**Program to print prime numbers between 1 to n**

```
/**
 * C program to print all prime numbers between 1 to n
 */
#include <stdio.h>
int main()
{
```

```

int i, j, n, isPrime; //isPrime is used as flag variable
/* Reads upper limit to print prime */
printf("Find prime numbers between 1 to : ");
scanf("%d", &n);
printf("\nAll prime numbers between 1 to %d are:\n", n);
/* Finds all Prime numbers between 1 to n */
for(i=2; i<=n; i++)
{
    /* Assume that the current number is Prime */
    isPrime = 1;
    /* Check if the current number i is prime or not */
    for(j=2; j<=i/2; j++)
    {
        /*
         * If i is divisible by any number other than 1 and self
         * then it is not prime number
         */
        if(i%j==0)
        {
            isPrime = 0;
            break;
        }
    }
    /* If the number is prime then print */
    if(isPrime==1)
    {
        printf("%d is Prime number\n", i);
    }
}
return 0;
}

```

**Note:** For checking whether a number is Prime or not we just need to check that the number should not be divisible by any number between 2 to  $n-1$ . Apart from that you can also check between 2 to  $n/2$ . Since any number more than  $(n/2)+1$  cannot be exactly divided by  $n$  except self  $n$ .

#### Output

Find prime numbers between 1 to : 100

All prime numbers between 1 to 100 are:

2 is Prime number

3 is Prime number

```
5 is Prime number
7 is Prime number
11 is Prime number
13 is Prime number
17 is Prime number
19 is Prime number
23 is Prime number
29 is Prime number
31 is Prime number
```

## Program to find GCD (HCF)

```
/**
 * C program to find HCF(Highest Common Factor) of two numbers
 */

#include <stdio.h>

int main()
{
    int i, num1, num2, min, hcf=1;

    /*
     * Reads two numbers from user
     */
    printf("Enter any two numbers to find HCF: ");
    scanf("%d %d", &num1, &num2);

    min = (num1<num2) ? num1 : num2;

    for(i=1; i<=min; i++)
    {
        /*
         * If i is factor of both number
         */
        if(num1%i==0 && num2%i==0)
        {
            hcf = i;
        }
    }
}
```

```

    printf("HCF of %d and %d = %d\n", num1, num2, hcf);
    return 0;
}

```

### Output

```

Enter any two numbers to find HCF: 12
30
HCF of 12 and 30 = 6

```

## Program to find LCM

```

/**
 * C program to find LCM of any two numbers
 */
#include <stdio.h>

int main()
{
    int i, num1, num2, max, lcm=1;
    /*
     * Reads two numbers from user
     */
    printf("Enter any two numbers to find LCM: ");
    scanf("%d %d", &num1, &num2);
    max = (num1>num2) ? num1 : num2;
    i = max;

    //Loop runs forever
    while(1)
    {
        /* If i is a multiple of both numbers */
        if(i%num1==0 && i%num2==0)
        {
            lcm = i;
            break;
        }
        i += max;
    }

    printf("\nLCM of %d and %d = %d\n", num1, num2, lcm);
    return 0;
}

```

```
}
```

Output

```
Enter any two numbers to find LCM: 12
30
LCM of 12 and 30 = 60
```

## Program to reverse the order of words in a given string

```
/**
 * C program to reverse order of words in a string
 */

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

int main()
{
    char string[100], reverse[100];
    int len, i, index, wordStart, wordEnd;

    printf("Enter any string: ");
    gets(string);

    len = strlen(string);
    index = 0;

    // Start checking of words from the end of string
    wordStart = len - 1;
    wordEnd = len - 1;

    while(wordStart > 0)
    {
        // If a word is found
        if(string[wordStart] == ' ')
        {
            // Add the word to the reverse string
            i = wordStart + 1;
            while(i <= wordEnd)
```

```

        {
            reverse[index] = string[i];

            i++;
            index++;
        }
        reverse[index++] = ' ';

        wordEnd = wordStart - 1;
    }

    wordStart--;
}

// Finally add the last word
for(i=0; i<=wordEnd; i++)
{
    reverse[index] = string[i];
    index++;
}
reverse[index] = '\\0'; // Adds a NULL character at the end of string

printf("Original string \\n%s\\n\\n", string);
printf("Reverse ordered words \\n%s", reverse);

return 0;
}

```

## Program to count frequency of digits in an integer

```

/**
 * C program to count frequency of digits in a given number
 */

#include <stdio.h>

```

```

#define BASE 10

int main()
{
    long long num, n;
    int i, lastDigit;
    int freq[BASE];

    printf("Enter any number: ");
    scanf("%lld", &num);

    // Initializes frequency array with 0
    for(i=0; i<BASE; i++)
    {
        freq[i] = 0;
    }

    n = num; //Copies the value of num to n

    while(n != 0)
    {
        // Gets the last digit
        lastDigit = n % 10;

        // Increments the frequency array
        freq[lastDigit]++;

        // Removes the last digit from n
        n /= 10;
    }

    printf("Frequency of each digit in %lld is: \n", num);
    for(i=0; i<BASE; i++)
    {
        printf("Frequency of %d = %d\n", i, freq[i]);
    }

    return 0;
}

```



```
}
```

```
Enter any number: 11203458760011
Frequency of each digit in 11203458760011 is:
Frequency of 0 = 3
Frequency of 1 = 4
Frequency of 2 = 1
Frequency of 3 = 1
Frequency of 4 = 1
Frequency of 5 = 1
Frequency of 6 = 1
Frequency of 7 = 1
Frequency of 8 = 1
Frequency of 9 = 0
```

## Program to print the given number pattern

### Example:

Input N: 5

Output:

```
  1
 11
101
1001
11111
```

```
/**
 * C program to print triangle 0, 1 number pattern
 */

#include <stdio.h>

int main()
{
    int i, j, N;

    printf("Enter N: ");
    scanf("%d", &N);
```

```

    for(i=1; i<=N; i++)
    {
        for(j=1; j<=i; j++)
        {
            if(i==1 || i==N || j==1 || j==i)
            {
                printf("1");
            }
            else
            {
                printf("0");
            }
        }

        printf("\n");
    }

    return 0;
}

```

Enter N: 5

```

1
11
101
1001
11111

```

## Program to print number pattern 2

```

1
123
12345
1234567
123456789
1234567
12345
123
1
1
123
12345
1234567
123456789
1234567
12345
123
1

```

```

/**
 * C program to print the given number pattern
 */
#include <stdio.h>

int main()
{
    int i, j, N;
    printf("Enter N: ");
    scanf("%d", &N);
    // Iterate through upper half triangle of the pattern
    for(i=1; i<=N; i++)
    {
        for(j=1; j<=(i * 2 - 1); j++)
        {
            printf("%d", j);
        }
        printf("\n");
    }
    // Iterate through lower half triangle of the pattern
    for(i=N-1; i>=1; i--)
    {
        for(j=1; j<=(i * 2 - 1); j++)
        {
            printf("%d", j);
        }
        printf("\n");
    }
    return 0;
}

```

```

Enter N: 5
1
123
12345
1234567
123456789
1234567
12345
123
1

```

## Program to print the given number pattern

### Example:

Input N:

Output:

1				
2	4			
3	6	9		
4	8	12	16	
5	10	15	20	25

```
/**
 * C program to print the given number pattern
 */

#include <stdio.h>

int main()
{
    int i, j, N;

    printf("Enter N: ");
    scanf("%d", &N);

    for(i=1; i<=N; i++)
    {
        for(j=i; j<=(i*i); j+= i)
        {
            printf("%-3d", j);

            printf("\n");
        }

        return 0;
    }
}
```

</stdio.h>

Enter N: 5

1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

