**A1. Check whether a given number can be expressed as the sum of two prime number**

**i/p**

Enter a positive integer: 34

**Output**

34 = 3 + 31

34 = 5 + 29

34 = 11 + 23

34 = 17 + 17

NoofWays = 4

NoofWays=-1

* **Code:**

#include <stdio.h>

int Prime(int n)

{

if (n <= 1)

return 0;

for (int i = 2; i \* i <= n; i++)

{

if (n % i == 0) return 0;

}

return 1;

}

int Sum(int n) {

int count = 0;

for (int i = 2; i <= n / 2; i++) {

if (Prime(i) && Prime(n - i)) {

printf("%d = %d + %d\n", n, i, n - i);

count++;

}

}

if (count == 0)

{

printf("NoofWays = -1\n");

} else

{

printf("NoofWays = %d\n", count);

}

}

int main()

{

int num;

printf("Enter a positive integer: ");

scanf("%d", &num);

if (num < 2)

{

printf("Input must be a positive integer greater than or equal to 2.\n");

return 0;

}

Sum(num);

return 0;

}

**A3. Write a program to find the n-th number made of prime digits**

Input Format:

First Line Of Input Contains T Number Of Test Cases

Second Line Of Input Contains an input Number N.

Output Format:

Print the Nth number of sequence and it should be only prime.

Constraints:

1<=T<=100

1<=N<=10000

Examples :

Input :

1

10

Output :

33

**Code :**

#include <stdio.h>

#define MAX 10000

int primeDigits[] = {2, 3, 5, 7};

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

int index = 0;

for (int i = 0; i < 4; i++) arr[index++] = primeDigits[i];

for (int length = 1; index < n; length++) {

for (int i = 0; i < index; i++) {

for (int j = 0; j < 4; j++) {

arr[index++] = arr[i] \* 10 + primeDigits[j];

if (index >= n)

return 0;

}

}

}

}

int main() {

int t, n, arr[MAX];

scanf("%d", &t);

Num(arr, MAX);

while (t--) {

scanf("%d", &n);

printf("%d\n", arr[n - 1]);

}

return 0;

}

**A4. rotate array every kth element**

1 2 3 4 5 6 7 8 9

k= 3

3 2 1 6 5 4 9 8 7

**Code:**

#include <stdio.h>

int rotate(int arr[], int n, int k) {

for (int i = 0; i < n; i += k) {

int a = i;

int b = i + k - 1;

if (b >= n) {

b = n - 1;

}

while (a < b) {

int temp = arr[a];

arr[a] = arr[b];

arr[b] = temp;

a++;

b--;

}

}

}

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

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

printf("%d ", arr[i]);

}

printf("\n");

}

int main() {

int n, k;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array:\n");

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

scanf("%d", &arr[i]);

}

printf("Enter the value of k: ");

scanf("%d", &k);

printf("Original Array: ");

Array(arr, n);

rotate(arr, n, k);

printf("Array after rotating every %dth element: ", k);

Array(arr, n);

return 0;

}

**5.)Write a c program to print alphabet triangle.**

Input: 5

Output:

A

ABA

ABCBA

ABCDCBA

ABCDEDCBA

**Code:**

#include <stdio.h>

int ATriangle(int n)

{

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

{

for (int j = 0; j < n - i - 1; j++) {

printf(" ");

}

for (int j = 0; j <= i; j++) {

printf("%c", 'A' + j);

}

for (int j = i - 1; j >= 0; j--) {

printf("%c", 'A' + j);

}

printf("\n");

}

}

int main() {

int n;

printf("Enter the number of rows: ");

scanf("%d", &n);

printf("\nAlphabet Triangle:\n");

ATriangle(n);

return 0;

}

A

ABA

ABCBA

ABCDCBA

ABCDEDCBA

**Code to print numbers**

**Number triangle**

#include<stdio.h>

int BTriangle(int n) {

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

for (int j = 0; j < n - i - 1; j++) {

printf(" ");

}

for (int j = 1; j <= i + 1; j++) {

printf("%d", j);

}

for (int j = i; j >= 1; j--) {

printf("%d", j);

}

printf("\n");

}

}

int main() {

int n;

printf("Enter the number of rows: ");

scanf("%d", &n);

printf("\nNumber Triangle:\n");

BTriangle(n);

return 0;

}

1

121

12321

1234321

123454321

**A2.Program to find nth number made of given four digits 1, 4,6 and 9 as the only digits..**

**Input Format:**

The First Line Of Input Contains T no of test cases

The Second Line Of Input Contains N as input taken.

**Output Format:**

Print the number of digits in the nth number .

**Constraints:**

1<=T<=100

1<=N<=100

Examples:

Input : 6

Output : 14

1,4 6, 9.11,14

Input : 21

Output : 111

**Code:**

#include <stdio.h>

void find\_nth\_number(int n)

{  
    int digits[] = {1, 4, 6, 9};  
    int result[100];

    int idx = 0;  
    while (n > 0) {  
  
        result[idx++] = digits[(n - 1) % 4];  
        n = (n - 1) / 4;    
    }  
    printf("The number is:");  
    for (int i = idx - 1; i >= 0; i--) {  
        printf("%d", result[i]);  
    }  
    printf("\n");  
}

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

    find\_nth\_number(n);

    return 0;  
}