Arnav Agrawal Lab 3 and Lab 4- 200905200

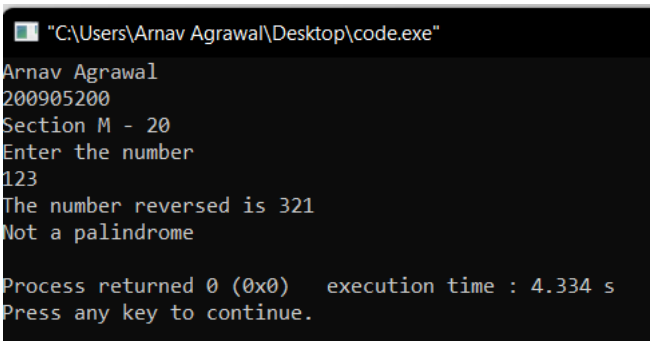
Lab 3

Question 1

Reverse a given number and check if it is a palindrome or not. (use while loop).

```
// Arnav Agrawal
// 200905200
// Lab 3
// Question 1
// Reverse a given number and check if it is a palindrome or not. (use while loop).
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int num;
    printf("Enter the number \n");
    scanf("%d", &num);
    int num_copy = num;
    int reverse = 0;
    int digit;
    while (num_copy > 0)
    {
        digit = num_copy % 10;
        reverse = reverse * 10 + digit;
        num_copy /= 10;
    }
    printf("The number reversed is %d \n", reverse);
    if (reverse == num)
    {
        printf("Palindrome \n");
    }
    else
        printf("Not a palindrome \n");

    return 0;
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 3
// Question 1
// Reverse a given number and check if it is a palindrome or not. (use while loop).
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int num;
    printf("Enter the number \n");
    scanf("%d", &num);
    int num_copy = num;
    int reverse = 0;
    int digit;
    while (num_copy > 0)
    {
        digit = num_copy % 10;
        reverse = reverse * 10 + digit;
        num_copy /= 10;
    }
    printf("The number reversed is %d \n", reverse);
    if (reverse == num)
    {
        printf("Palindrome \n");
    }
    else
        printf("Not a palindrome \n");

    return 0;
}
```



Question 2

Generate prime numbers between 2 given limits.(use while loop)

```c
// Arnav Agrawal
// 200905200
// Lab 3
// Question 2
// Generate primenumbers between 2 given limits.(use while loop)
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int a, b;
    printf("Enter the limits\n");
    scanf("%d %d", &a, &b);
    int start = a;
    while (start <= b)
    {
```

```
        int k = 2;
        int flag = 1;
        while (k < start)
        {
            if (start % k == 0)
            {
                flag = 0;
                break;
            }
            k++;
        }
        if (flag == 1 && start != 1)
        {
            printf("%d\t", start);
        }
        start++;
    }
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 3
// Question 2
// Generate prime numbers between 2 given limits.(use while loop)
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int a, b;
    printf("Enter the limits\n");
    scanf("%d %d", &a, &b);
    int start = a;
    while (start <= b)
    {
        int k = 2;
        int flag = 1;
        while (k < start)
        {
            if (start % k == 0)
            {
                flag = 0;
                break;
            }
            k++;
        }
        if (flag == 1 && start != 1)
        {
            printf("%d\t", start);
        }
        start++;
    }
}
```

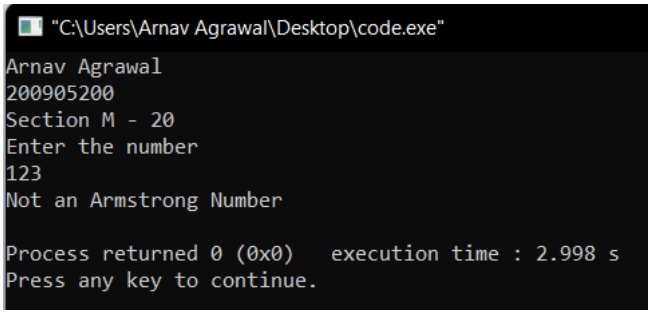
```
"C:\Users\Arnav Agrawal\Desktop\code.exe"
Arnav Agrawal
200905200
Section M - 20
Enter the limits
1 100
2       3       5       7       11      13      17      19      23      29      31      37      41      43      47
        53      59      61      67      71      73      79      83      89      97
Process returned 0 (0x0)   execution time : 8.668 s
Press any key to continue.
```

Question 3

Check if the sum of the cubes of all digits of an inputted number equals the number itself (Armstrong Number). (use while loop)

```
// Arnav Agrawal
// 200905200
// Lab 3
// Question 3
// Check if the sum of the cubes of all digits of an inputted number equals the number itself (Armstrong Number). (use while loop)
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int num;
    printf("Enter the number \n");
    scanf("%d", &num);
    int num_copy = num;
    int arms = 0;
    int digit;
    while (num_copy > 0)
    {
        digit = num_copy % 10;
        arms = arms + digit * digit * digit;
        num_copy /= 10;
    }
    if (arms == num)
    {
        printf("Armstrong Number \n");
    }
    else
        printf("Not an Armstrong Number \n");

    return 0;
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 3
// Question 3
// Check if the sum of the cubes of all digits of an inputted number
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int num;
    printf("Enter the number \n");
    scanf("%d", &num);
    int num_copy = num;
    int arms = 0;
    int digit;
    while (num_copy > 0)
    {
        digit = num_copy % 10;
        arms = arms + digit * digit * digit;
        num_copy /= 10;
    }
    if (arms == num)
    {
        printf("Armstrong Number \n");
    }
    else
        printf("Not an Armstrong Number \n");

    return 0;
}
```



Question 4

Write a program using do-while loop to read the numbers until -1 is encountered. Also count the number of prime numbers and composite numbers entered by user. [Hint: 1 is neither prime nor composite]

```
// Arnav Agrawal
// 200905200
// Lab 3
// Question 4
// Write a program using do-while loop to read the numbers until -1 is encountered. Also count the number of prime numbers and composi
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int num = 0;
    int total = 0;
    int prime = 0;
    while (num != -1)
    {
        printf("Enter a number \n");
        scanf("%d", &num);
        int k = 2;
        int flag = 1;
        while (k < num)
        {
            if (num % k == 0)
            {
                flag = 0;
                break;
            }
            k++;
        }
        if (flag == 1 && num > 1)
        {
            prime++;
        }
        total++;
    }
    printf("Total numbers entered: %d\n", total-1);
    printf("Total prime numbers entered: %d\n", prime);
    printf("Total composite numbers entered: %d\n", total - prime -1);

    return 0;
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 3
// Question 4
// Write a program using do-while loop to read the numbers until -1 is encou
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int num = 0;
    int total = 0;
    int prime = 0;
    while (num != -1)
    {
        printf("Enter a number \n");
        scanf("%d", &num);
        int k = 2;
        int flag = 1;
        while (k < num)
        {
            if (num % k == 0)
            {
                flag = 0;
                break;
            }
            k++;
        }
        if (flag == 1 && num > 1)
        {
            prime++;
        }
        total++;
    }
    printf("Total numbers entered: %d\n", total-1);
    printf("Total prime numbers entered: %d\n", prime);
    printf("Total composite numbers entered: %d\n", total - prime -1);

    return 0;
}
```

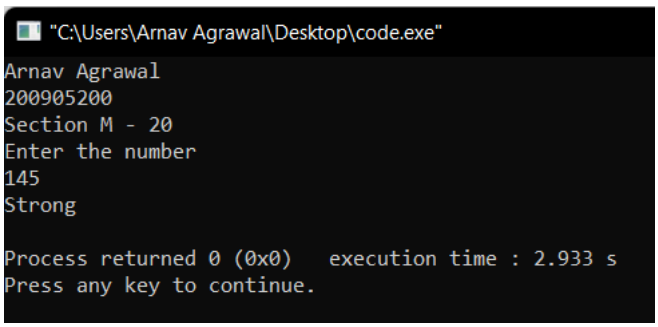
```
"C:\Users\Arnav Agrawal\Desktop\code.exe"
Arnav Agrawal
200905200
Section M - 20
Enter a number
10
Enter a number
12
Enter a number
11
Enter a number
13
Enter a number
17
Enter a number
-1
Total numbers entered: 5
Total prime numbers entered: 3
Total composite numbers entered: 2

Process returned 0 (0x0)   execution time : 10.721 s
Press any key to continue.
```

Question 5

Check whether the given number is strong or not.

```
// Arnav Agrawal
// 200905200
// Lab 3
// Question 5
// Check whether the given number is strong or not.
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int num;
    printf("Enter the number \n");
    scanf("%d", &num);
    int num_copy = num;
    int new_num = 0;
    int digit;
    while (num_copy > 0)
    {
        digit = num_copy % 10;
        int i = 1;
        int fact=1;
        while (i <= digit)
        {
            fact *= i;
            i++;
        }
        new_num += fact;
        num_copy /=10;
    }
    if (new_num == num)
    {
        printf("Strong \n");
    }
    else
        printf("Not strong \n");
    return 0;
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 3
// Question 5
// Check whether the given number is strong or not.
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int num;
    printf("Enter the number \n");
    scanf("%d", &num);
    int num_copy = num;
    int new_num = 0;
    int digit;
    while (num_copy > 0)
    {
        digit = num_copy % 10;
        int i = 1;
        int fact=1;
        while (i <= digit)
        {
            fact *= i;
            i++;
        }
        new_num += fact;
        num_copy /=10;
    }
    if (new_num == num)
    {
        printf("Strong \n");
    }
    else
        printf("Not strong \n");
    return 0;
}
```



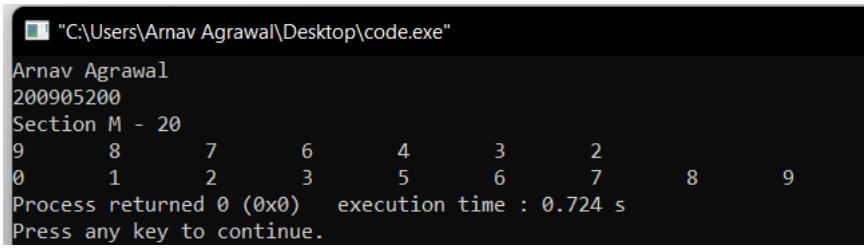
Question 6

Write a program to demonstrate use of break and continue statements in while and do-while loops.

```c
// Arnav Agrawal
// 200905200
// Lab 3
// Question 6
// Write a program to demonstrate use of break and continue statements in while and do-while loops.
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
```

```
    int i = 9;
    while (i >= 0)
    {
        if (i == 5)
        {
            i--;
            continue;
        }
        printf("%d\t", i);
        i--;
        if (i == 1)
            break;
    }
    printf("\n");
    int j = 0;
    do
    {
        if (j == 4)
        {
            j++;
            continue;
        }
        printf("%d\t", j);
        j++;
        if (j == 10)
            break;
    } while (j < 10);

    return 0;
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 3
// Question 6
// Write a program to demonstrate use of break and continue
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int i = 9;
    while (i >= 0)
    {
        if (i == 5)
        {
            i--;
            continue;
        }
        printf("%d\t", i);
        i--;
        if (i == 1)
            break;
    }
    printf("\n");
    int j = 0;
    do
    {
        if (j == 4)
        {
            j++;
            continue;
        }
        printf("%d\t", j);
        j++;
        if (j == 10)
            break;
    } while (j < 10);

    return 0;
}
```



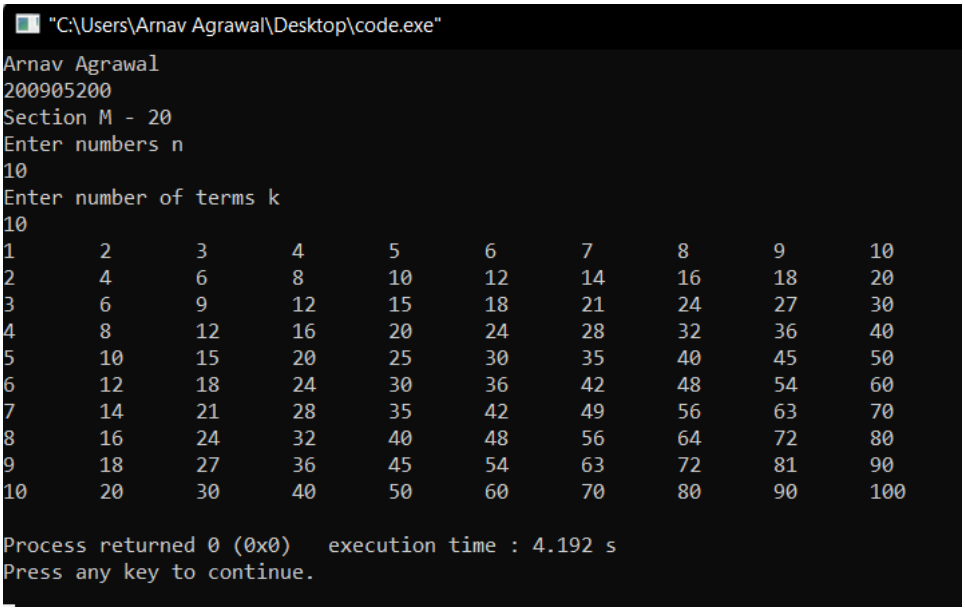
Lab 4

Question 1

Generate the multiplication table for ‘n ’numbers up to ‘k’ terms (using nested for loops).

```
// Arnav Agrawal
// 200905200
// Lab 4
// Question 1
// Generate the multiplication table for ‘n’ numbers up to ‘k’ terms (using nestedfor loops)
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n, k;
    printf("Enter numbers n\n");
    scanf("%d", &n);
    printf("Enter number of terms k\n");
    scanf("%d", &k);
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= k; j++)
        {
            printf("%d \t", (i * j));
        }
        printf("\n");
    }
    return 0;
}
```

```
// Arnav Agrawal
// 200905200
// Lab 4
// Question 1
// Generate the multiplication table for ‘n’ numbers up to ‘k’ terms
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n, k;
    printf("Enter numbers n\n");
    scanf("%d", &n);
    printf("Enter number of terms k\n");
    scanf("%d", &k);
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= k; j++)
        {
            printf("%d \t", (i * j));
        }
        printf("\n");
    }
    return 0;
}
```

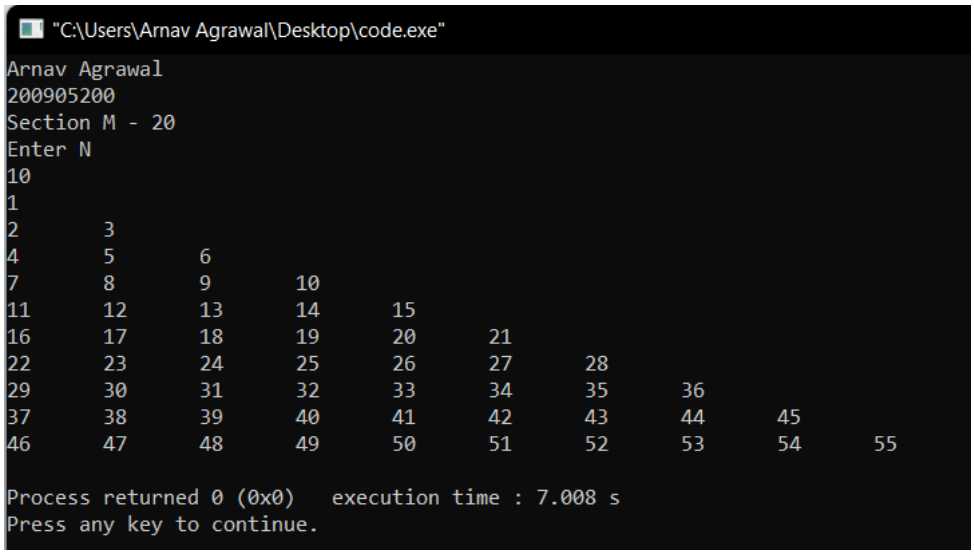


Question 2

Generate Floyd's triangle using natural numbers for a given limit N. (using for loops)

```
// Arnav Agrawal
// 200905200
// Lab 4
// Question 2
// Generate Floyd's triangle using natural numbers for a given limit N
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n;
    printf("Enter N\n");
    scanf("%d", &n);
    int c = 1;
    for (int j = 1; j <= n; j++)
    {
        for (int i = 0; i < j; i++)
        {
            printf("%d\t", c);
            c++;
        }
        printf("\n");
    }

    return 0;
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 4
// Question 2
// Generate Floyd's triangle using natural numbers for a given limit N
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n;
    printf("Enter N\n");
    scanf("%d", &n);
    int c = 1;
    for (int j = 1; j <= n; j++)
    {
        for (int i = 0; i < j; i++)
        {
            printf("%d\t", c);
            c++;
        }
        printf("\n");
    }

    return 0;
}
```



Question 3

Evaluate the sine series, sin(x)= x-x3/3! + x5/5!–x7/7!+ … to n terms.

```c
// Arnav Agrawal
// 200905200
// Lab 4
// Question 3
// Evaluate the sine series, sin(x)= x-x3/3! + x5/5!-x7/7!+ ......... to n terms.
#include <stdio.h>
#include <math.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n;
    printf("Enter N\n");
    scanf("%d", &n);
    float x;
```

```
    printf("Enter the angle in radians\n");
    scanf("%f", &x);
    float curr_term, sum;
    curr_term = x;
    sum = x;
    for (int i = 1; i <= n; i++)
    {
        curr_term = curr_term * (((-1) * pow(x, 2)) / (2 * i * (2 * (i) + 1)));
        sum += curr_term;
    }
    printf("Library value of Sin(%f) =  %f \n", x, sin(x));
    printf("Whilst we get, Sin (%f) = %f", x, sum);

    return 0;
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 4
// Question 3
// Evaluate the sine series, sin(x)= x-x3/3! + x5/5!-x7/7!+ ......... to n terms.
#include <stdio.h>
#include <math.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n;
    printf("Enter N\n");
    scanf("%d", &n);
    float x;
    printf("Enter the angle in radians\n");
    scanf("%f", &x);
    float curr_term, sum;
    curr_term = x;
    sum = x;
    for (int i = 1; i <= n; i++)
    {
        curr_term = curr_term * (((-1) * pow(x, 2)) / (2 * i * (2 * (i) + 1)));
        sum += curr_term;
    }
    printf("Library value of Sin(%f) =  %f \n", x, sin(x));
    printf("Whilst we get, Sin (%f) = %f", x, sum);

    return 0;
}
```

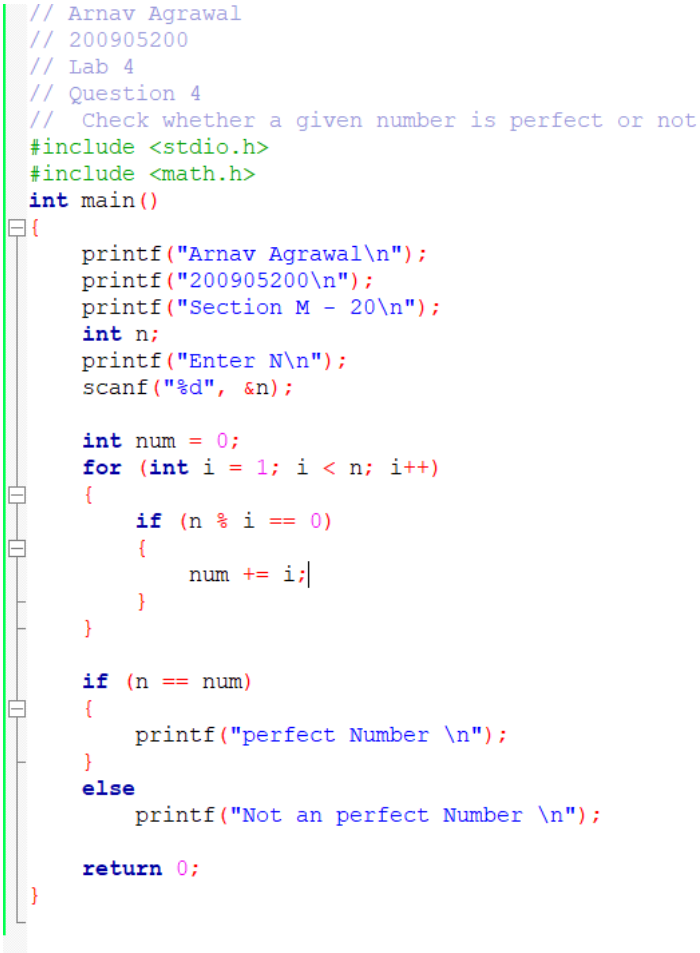
```
"C:\Users\Arnav Agrawal\Desktop\code.exe"
Arnav Agrawal
200905200
Section M - 20
Enter N
10
Enter the angle in radians
1.6
Library value of Sin(1.600000) =  0.999574
Whilst we get, Sin (1.600000) = 0.999574
Process returned 0 (0x0)   execution time : 12.734 s
Press any key to continue.
```

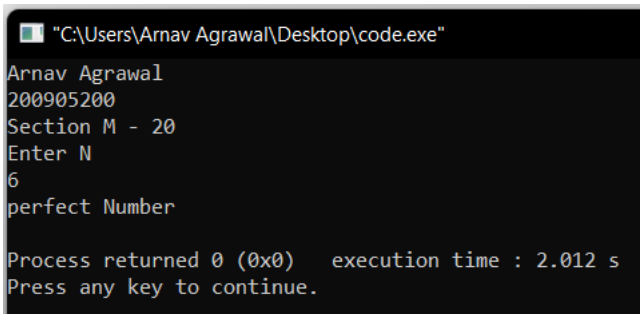
Question 4

Check whether a given number is perfect or not.

```
// Arnav Agrawal
// 200905200
// Lab 4
```

```
// Question 4
//  Check whether a given number is perfect or not
#include <stdio.h>
#include <math.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n;
    printf("Enter N\n");
    scanf("%d", &n);

    int num = 0;
    for (int i = 1; i < n; i++)
    {
        if (n % i == 0)
        {
            num += i;
        }
    }

    if (n == num)
    {
        printf("perfect Number \n");
    }
    else
        printf("Not an perfect Number \n");

    return 0;
}
```





Question 5

Find out the generic root of any number.

```
// Arnav Agrawal
// 200905200
// Lab 4
// Question 5
// Find out the generic root of any number.
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n;
    printf("Enter N\n");
    scanf("%d", &n);
    int sum = 0;
    for (; n > 0;)
    {
        int d = n % 10;
        sum += d;
        n /= 10;
    }

    if (sum > 9)
    {
        int sum_copy = sum;
        sum = 0;
        for (; sum_copy > 0;)
        {
            int d = sum_copy % 10;
            sum += d;
            sum_copy /= 10;
        }
        printf("The generic root is %d\n", sum);
    }
    else
    {
        printf("The generic root is %d\n", sum);
    }
    return 0;
}
```

```c
// Arnav Agrawal
// 200905200
// Lab 4
// Question 5
// Find out the generic root of any number.
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int n;
    printf("Enter N\n");
    scanf("%d", &n);
    int sum = 0;
    for (; n > 0;)
    {
        int d = n % 10;
        sum += d;
        n /= 10;
    }

    if (sum > 9)
    {
        int sum_copy = sum;
        sum = 0;
        for (; sum_copy > 0;)
        {
            int d = sum_copy % 10;
            sum += d;
            sum_copy /= 10;
        }
        printf("The generic root is %d\n", sum);
    }
    else
    {
        printf("The generic root is %d\n", sum);
    }
    return 0;
}
```

```
"C:\Users\Arnav Agrawal\Desktop\code.exe"
Arnav Agrawal
200905200
Section M - 20
Enter N
99
The generic root is 9

Process returned 0 (0x0)   execution time : 2.005 s
Press any key to continue.
```

Question 6

Write a program to demonstrate use of break and continue statements in for loop.

```c
// Arnav Agrawal
// 200905200
// Lab 4
// Question 6
// Write a program to demonstrate use of break and continue statements in for loop.
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int i = 0, j = 0;
    for (int i = 1; i < 100; i++)
```

```
    {
        if (i % 2 == 0)
        {
            continue;
        }
        if (i == 21)
        {
            break;
        }

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

```c
// Arnav Agrawal
// 200905200
// Lab 4
// Question 6
// Write a program to demonstrate use of break and continue statements in for loop.
#include <stdio.h>
int main()
{
    printf("Arnav Agrawal\n");
    printf("200905200\n");
    printf("Section M - 20\n");
    int i = 0, j = 0;
    for (int i = 1; i < 100; i++)
    {
        if (i % 2 == 0)
        {
            continue;
        }
        if (i == 21)
        {
            break;
        }

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