Day : Loops and Iterations (5-8-2025)

1. Write a program to print numbers from 1 to 100.

IPO

Input: No input is required.

Process: Use a loop to print numbers starting from 1 up to 100.

Output: The numbers from 1 to 100 displayed sequentially.

Program

```c
#include <stdio.h>

int main()
{
    int i;

    for (i = 1; i <= 100; i++)
    {
        printf("%d", i);
    }

    return 0;
}
```

Output

2. Write a program to print even numbers from 1 to 50.

IPO

Input: No input is required.

Process: Loop through numbers from 1 to 50 and check if each is divisible by 2; print only even ones.

Output: Even numbers between 1 and 50.

Program

```c
#include <stdio.h>


int main()
{
   int i;


   for (i = 1; i <= 50; i++)
   {
     if (i % 2 == 0)
     {
        printf("%d", i);
     }
   }


   return 0;
}
```

Output

3. Write a program to find the factorial of a number.

IPO

Input: The user enters a positive integer.

Process: Multiply the numbers from 1 to the entered number to get the factorial.

Output: The factorial value of the entered number.

Program

```c
#include <stdio.h>
void main()
{
    int n, i, fact = 1;
    scanf("%d", &n);
    for(i = 1; i <= n; i++)
    {
        fact = fact * i;
    }
    printf("%d", fact);
}
```

Output

4. Write a program to calculate the sum of digits of a number.

IPO

Input: The user enters an integer.

Process: Extract each digit of the number and add them together.

Output: The sum of all digits in the number.

Program

```c
#include <stdio.h>
void main()
{
```
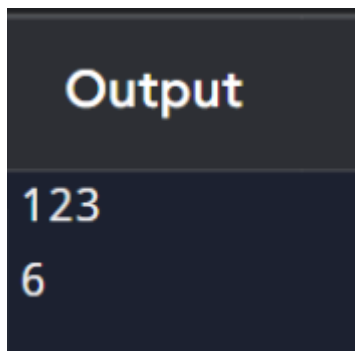
```c
    int i, sum = 0, r, n;

    scanf("%d", &n);

    while(n > 0)

    {

        r = n % 10;

        sum = sum + r;

        n = n / 10;

    }

    printf("%d", sum);

}
```

Output

5. Write a program to reverse a number.

IPO

Input: The user enters a number.

Process: Reverse the digits of the entered number.

Output: The reversed number.

Program

```c
#include <stdio.h>

void main()

{

    int n, r, rev = 0;

    scanf("%d", &n);

    while(n > 0)

    {
```
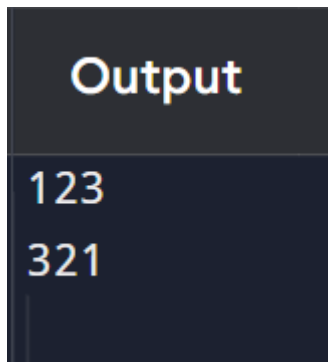
```
    r = n % 10;

    rev = rev * 10 + r;

    n = n / 10;

  }

  printf("%d", rev);
```

output

6. Write a program to check whether a number is a palindrome.

IPO

Input: The user enters a number.

Process: Reverse the number and compare it with the original to check if they are the same.

Output: A message stating whether the number is a palindrome or not.

Program

```
#include <stdio.h>

void main()

{

  int n, r, rev = 0, sum = 0, c = 100, on;

  scanf("%d", &on);

  n = on;

  while(n > 0)

  {

    r = n % 10;

    rev = rev + r * c;

    n = n / 10;

    c = c / 10;
```
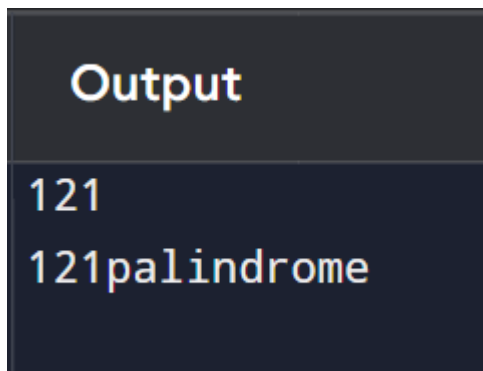
```
    }
    printf("%d", rev);
    if(rev == on)
        printf("palindrome");
    else
        printf("not a palindrome");
}
```

Output


```
Output

121
121palindrome
```

7. Write a program to print multiplication table of a number.

IPO

Input: The user enters a number.

Process: Multiply the number by values from 1 to 10 and display results.

Output: The multiplication table for the entered number.

Program

```c
#include <stdio.h>
void main()
{
    int n, i;
    scanf("%d", &n);
    for(i = 1; i <= n; i++)
    {
        printf("%d * %d = %d\n", n, i, n * i);
    }
```

```

10
10 * 1 = 10
10 * 2 = 20
10 * 3 = 30
10 * 4 = 40
10 * 5 = 50
10 * 6 = 60
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100
```

8. Write a program to count the number of digits in a number.

IPO

 Input: The user enters a number.

 Process: Divide the number by 10 repeatedly and count how many times this is done until it becomes zero.

 Output: The total number of digits in the number.

Program

#include <stdio.h>

int main()

{

```c
    int num, count = 0;

    printf("Enter a number: ");
    scanf("%d", &num);

    // If number is 0, it has 1 digit
    if (num == 0)
    {
        count = 1;
    }
    else
    {
        // Make number positive if it's negative
        if (num < 0)
        {
            num = -num;
        }

        while (num > 0)
        {
            num = num / 10; // remove last digit
            count++;        // increase digit count
        }
    }

    printf("Number of digits: %d\n", count);

    return 0;
}
```
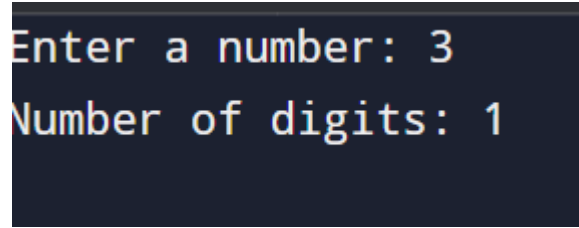Output

```
Enter a number: 3
Number of digits: 1
```

9. Write a program to print the Fibonacci series up to n terms.

IPO

Input: The user enters the number of terms (n).

Process: Calculate and display the first n terms of the Fibonacci sequence.
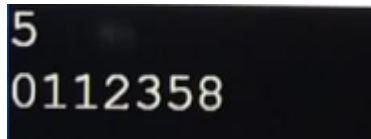
Output: The Fibonacci series up to n terms.

Program

```c
#include <stdio.h>
void main()
{
    int f = 0, s = 1, t, i, n;
    scanf("%d", &n);
    printf("%d %d", f, s);
    for(i = 1; i < n; i++)
    {
        t = f + s;
        f = s;
        s = t;
        printf("%d", t);
    }
}
```

Output

```
5
0112358
```

10. Write a program to calculate the sum of the first n natural numbers.

IPO

Input: The user enters a positive integer n.

Process: Add numbers from 1 to n.
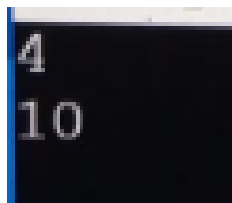
Output: The sum of the first n natural numbers.

Program

```c
#include <stdio.h>
void main()
{
    int n, sum = 0, i;
    scanf("%d", &n);
    for(i = 1; i <= n; i++)
    {
        sum = sum + i;
    }
    printf("%d", sum);
}
```

Output

```
4
10
```