Sulakshi Sewmini Samarakoon
29878

Practical 05

1)

- While

```
#include <stdio.h>

int main() {
    int number = 0;

    while (number <= 100) {
        printf("%d ", number);
        number++;
    }

    return 0;
}
```

- Do while

```
#include <stdio.h>

int main() {
    int number = 0;

    do {
        printf("%d ", number);
        number++;
    } while (number <= 100);

    return 0;
}
```

- For

```
#include <stdio.h>
```

```
int main() {
    for (int number = 0; number <= 100; number++) {
        printf("%d ", number);
    }

    return 0;
}
```

2)

```
#include <stdio.h>

int main() {
    int marks[10];
    int total = 0;

    printf("Enter 10 marks:\n");
    for (int i = 0; i < 10; i++) {
        scanf("%d", &marks[i]);
        total += marks[i];
    }

    float average = (float)total / 10;

    printf("Total: %d\n", total);
    printf("Average: %.2f\n", average);

    if (average < 50) {
        printf("Fail!\n");
    } else {
        printf("Pass!\n");
    }

    return 0;
}
```

3)

```
#include <stdio.h>

int main() {
    int number;
    int factorial = 1;

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

    if (number < 0) {
        printf("Factorial is not defined for negative numbers.\n");
    } else {
        for (int i = 1; i <= number; i++) {
            factorial *= i;
        }

        printf("Factorial of %d is %d\n", number, factorial);
    }

    return 0;
}
```

4)

```
#include <stdio.h>

int main() {
    int number, sum = 0;

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

    int remainder;
    while (number > 0) {
        remainder = number % 10;
        sum += remainder;
        number /= 10;
    }
```

```
    printf("Sum of digits: %d\n", sum);

    return 0;
}
```

5)

```
#include <stdio.h>

int main() {
    int number, reversedNumber = 0, remainder;

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

    do {
        remainder = number % 10;
        reversedNumber = reversedNumber * 10 + remainder;
        number = number / 10;
    } while (number != 0);

    printf("Reversed number: %d\n", reversedNumber);

    return 0;
}
```

6)

```
#include <stdio.h>

int main() {
    int base, exponent, result = 1;

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

    printf("Enter the exponent: ");
```

```
    scanf("%d", &exponent);

    int i;
    for (i = 0; i < exponent; i++) {
        result *= base;
    }

    printf("%d raised to the power %d is: %d\n", base, exponent, result);

    return 0;
}
```

7)

```
#include <stdio.h>

int main() {
    int n = 10;
    int fib[n];
    int i;

    fib[0] = 0;
    fib[1] = 1;

    for (i = 2; i < n; i++) {
        fib[i] = fib[i-1] + fib[i-2];
    }

    printf("The first 10 numbers of the Fibonacci sequence are:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", fib[i]);
    }
    printf("\n");

    return 0;
}
```

8)

```
#include <stdio.h>

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

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

    originalNumber = number;

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

    originalNumber = number;

    while (originalNumber != 0) {
        remainder = originalNumber % 10;
        int power = 1;
        for (int i = 1; i <= n; ++i) {
            power *= remainder;
        }
        result += power;
        originalNumber /= 10;
    }

    if (result == number)
        printf("%d is an Armstrong number.\n", number);
    else
        printf("%d is not an Armstrong number.\n", number);

    return 0;
}
```

9)

```
#include <stdio.h>

int main() {
    char letter;

    printf("ASCII values for letters A to Z:\n");

    for (letter = 'A'; letter <= 'Z'; ++letter) {
        printf("%c: %d\n", letter, letter);
```

```
    }

    return 0;
}
```

10)

```
#include <stdio.h>

int main() {
    int rows = 5;  // number of rows in the pattern
    int i, j;

    for (i = 1; i <= rows; ++i) {
        for (j = 1; j <= i; ++j) {
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```