Task-3

Добавлены функции:

```cpp
int Area(int length, int width)
{
    int area = length * width;
    return area;
}

int Perimetr(int length, int width)
{
    int perimetr = 2 * (length + width);
    return perimetr;
}

void DrawRectangle(int length, int width, char symbol)
{
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < width; j++)
        {
            cout << symbol;
        }
        cout << endl;
    }
}
```

```cpp
void Fibbonacci(int number)
{
    int fibbonacci, firstNumberFibonacci = 1, secondNumberFibonacci = 0;

    for (int i = 1; i <= number; i++)
    {
        fibbonacci = firstNumberFibonacci + secondNumberFibonacci;
        firstNumberFibonacci = secondNumberFibonacci;
        secondNumberFibonacci = fibbonacci;
        cout << fibbonacci << " ";
    }

    cout << endl;
}

long long Factorial(int number)
{
    long long factorial = 1;
    for (int i = 1; i <= number; i++)
    {
        factorial *= i;
    }

    return factorial;
}
```

```
string Prime(int number)
{
    for (int i = 2; i <= sqrt(number); i++)
    {
        if (number % i == 0)
        {
            return " не является простым числом";
        }
    }

    return "простое число";
}
int FindNOD(int firstNumber, int secondNumber)
{
    while (firstNumber != secondNumber)
    {
        if (firstNumber > secondNumber)
        {
            firstNumber = firstNumber - secondNumber;
        }
        else
        {
            secondNumber = secondNumber - firstNumber;
        }
    }

    return firstNumber;
}
```

```
string ConvertToBinary(long long number)
{
    string convertedNumberToBinary = "";
    while (number > 0) {
        convertedNumberToBinary += char(number % 2 + 48);
        number /= 2;
    }
    reverse(convertedNumberToBinary.begin(), convertedNumberToBinary.end());

    return convertedNumberToBinary;
}
```

Добавлены функции-меню

```
void AreaAndPerimetr();
void FibbonachiAndFactorial();
void SimpleAndNOD();
void BinaryConverter();
```