Jonas Carlsson
Joca1604@student.miun.se

Labb3 Programspråksteori

In this labb a program was created in C++ which would calculate exponential multiplications in Compile Time. This was achieved by creating templates of the POW function.

The base Template takes in the parameter exp, corresponding to the exponent in the calculation.

The struct consists of a function called by the operator (). The function takes a double bas as input, corresponding to the base value of the calculation. The function works by iterating down exp recursively by creating a new template object with exp-1. The function returns the base value multiplied by the returned value of the recursive function.

```
template <int exp, class Enable = void>
struct POW
{
    double operator()(double bas){
        POW<exp - 1> nextExp;
        return nextExp(bas) * bas;
    }
};
```

Once the exponent reaches 0 in the recursive function a separate version of the template is created.

Since a value with the exponent 0 always is 1 this function now returns 1 and stops the recursion.

```
template <>
struct POW<0>
{
    double operator()(double bas){
        return 1;
    }
};
```

A special case was also created in order to handle negative exponents. This works by using typename enable_if_t to use an if case in compiletime. It checks if the exponent is less than 0 before being able to run. If the enable_if is false the base temlate will be used. If the exponent is negative the function will call on the base template in recursion by switching polarity of the exponent. This template will return 1/base template calculation.

```
template <int exp>
struct POW<exp, typename std::enable_if_t<exp < 0> >
{
    double operator()(double bas){
        POW<-exp> nextExp;
        return 1/nextExp(bas);
    }
};
```