**LAB 13**

**Templates**

**18th June, 2021**

**TASK 1**

Write a program which is capable of calculating Raise to Power, Natural Logarithm, Square Root, Sin, Cos, Tan of any data type. For the mathematical calculations you are allowed to use C++ ‘cmath’ library. Your program should consist of generic function (Function templates) for each operation. In main () perform calculations for int, float, double type variables. Pass the operands to desired function which returns the result after performing appropriate operation. Print the result in main (). Your program must be user friendly and menu driven.

**TASK 2**

Just like arrays, vectors use contiguous storage locations for their elements, which means that their elements can also be accessed using offsets on regular pointers to its elements, and just as efficiently as in arrays. But unlike arrays, their size can change dynamically, with their storage being handled automatically.

Your task is to write a template for the Vector class that supports at the minimum following operations over different types (int, float, double, char, string) of Vectors created from the template:

* constructor
* destructor
* push back (Add element at the end)

Adds a new element at the end of the vector, after its current last element. The content of val is copied (or moved) to the new element. This effectively increases the storage size by one, which causes an automatic reallocation of the allocated storage space if -and only if- the new vector size surpasses the current vector capacity.

* pop back (Delete last element)

Removes the last element in the vector, effectively reducing the storage size by one. This destroys the removed element.

* operator [] (Access element)
* erase (Erase elements)

Removes from the vector either a single element (position) or a range of elements ([first,last]).This effectively reduces the storage size by the number of elements removed, which are destroyed.

* clear (Clear content)

Removes all elements from the vector (which are destroyed), leaving the storage with a size of 0.

* empty (Test whether vector is empty)

Returns whether the vector is empty (i.e. whether its size is 0).

This function does not modify the container in any way.

* resize (Change size)

Resizes the container so that it contains n elements. If n is smaller than the current container size, the content is reduced to its first n elements, removing those beyond (and destroying them).

If n is greater than the current container size, the content is expanded by inserting at the end as many elements as needed to reach a size of n. If val is specified, the new elements are initialized as copies of val, otherwise, they are initialized with a default value.

If n is also greater than the current container capacity, an automatic reallocation of the allocated storage space takes place.

Notice that this function changes the actual content of the container by inserting or erasing elements from it. to input and then display the numbers in tabular format.