Zaawansowane techniki programowania obiektowego w C++

Zestaw nr 3

1. Podaj implementacje funkcji **max** dla różnego typu argumentów. Skorzystaj z możliwości przeciążania i specjalizacji szablonów funkcji:

```
max(T a, T b)
max(T *a, T *b)
max(T *data,size_t n)
max(char *a,char *a)
max(const char *a,const char *a)
max(char *a,const char *a)
```

2. Na przykładzie szablonu funkcji **convert** pokazać, że jest możliwe zdefiniowanie dwóch funkcji o tej samej nazwie i argumentach wywołania, różniących się tylko zwracanym typem.

3. W przeciwieństwie do szablonów funkcji, szablony klas nie mogą być przeciążane, a jedynie specjalizowane. Oznacza to, że w programie może istnieć tylko jeden szablon podstawowy o danej nazwie. Proszę zaimplementować specjalizację częściową, która jest dozwolona tylko dla szablonów klas, dla podanych podzbiorów parametrów:

```
Stack<T,666>
Stack<T*,N>
Stack<double ,N>
Stack<int *,N>
Stack<double,666>
Stack<double *,44>
```

4. Zaimplementować szablon klasy **Stack wraz z** jej specjalizacjami tak aby „działała” ze zwykłymi tablicami oraz kontenerami STL:

```
Stack<int,100>              s_default ;
Stack<int,0,std::vector<int> >  s_container;
```

5. Zapoznaj się z rodziną funktorów porównujących, do której należą np. **less<>(), greater<>(), less_equal<>(), greater_equal<>()**. Przypomnij sobie pojemnik **std::set**. Następnie napisz programik korzystający ze zbioru liczb posortowanego odwrotnie (czyli malejąco).