

## STL ciąg dalszy

### Zadanie 1

Utwórz obiekt `Record` służący do przechowywania pewnych danych o osobach np.:

```
class Record
{
public:
    std::string mName;
    std::string mPhone;
    int      mAge;
};
```

### Zadanie 2

Dopisz do powyższego obiektu konstruktory domyślny `Record()` i jednoargumentowy `Record(const char c[])` który inicjalizuje atrybut `mName` wartością argumentu `c`.

### Zadanie 3

W funkcji głównej programu zadeklaruj tablicę obiektów typu `Record` korzystając ze standardowego kontenera `vector`.

### Zadanie 4

Wypełnij tablicę pewną ilością danych tak aby kilka razy powtórzyły się osoby o tym samym nazwisku np. „Nowak” i występowały inne nazwiska zaczynające się na literę N.

### Zadanie 5

Skopiuj powyższą tablicę do innej tablicy tymczasowej tego samego typu korzystając z algorytmu `copy( itr_beg, itr_end, dest_itr_beg)` gdzie `itr_beg` i `itr_end` określając zakres źródła do skopiowania a `dest_itr_beg` określa początek kontenera gdzie mają być wstawiane elementy.

### Zadanie 6

Wydrukuj zawartość tablicy na ekran korzystając z algorytmu `for_each(...)`. Ostatnim argumentem tego algorytmu jest funkcja lub obiekt funkcyjny dokonujący operacji na każdym z obiektów kolekcji znajdującym się pomiędzy podanymi iteratorami. Sprawdź działanie tego algorytmu korzystając zarówno z funkcji jak i z obiektu funkcyjnego. Przykładowy obiekt funkcyjny wygląda tak:

```
class Print
{
public:
    void operator () (const Record& rec)
    {
        // instrukcje które będą wykonywane
    }
};
```

### Zadanie 7

Wydrukuj zawartość tablicy do pliku korzystając ze strumienia `ofstream` i zmodyfikowanego obiektu funkcyjnego, który będzie dodatkowo przechowywał strumień (konieczny konstruktor!).

### Zadanie 8

Znajdź w tablicy wszystkie wystąpienia osób z nazwiskiem „Nowak”. W tym celu posortuj (algorytm `sort`) tablicę a następnie skorzystaj z algorytmu `equal_range( itr_begin, itr_end, val)` który zwraca obiekt typu `pair<iterator,iterator>` a jego atrybuty `first` i `second` tej pary odpowiednio przechowują iteratory do pierwszego i znajdującego się za ostatnim obiektu równego `val`. Aby można było skorzystać z tego algorytmu niezbędne jest dopisanie operatora `<` do obiektu `Record`. Wynik wydrukuj na ekran korzystając z `for_each`.

### Zadanie 9

Znajdź wszystkie elementy tablicy dla których nazwisko osoby zaczyna się na literę N. Skorzystaj z algorytmu `lower_bound( itr_begin, itr_end, val)` który zwraca iterator do pierwszego elementu o wartości nie mniejszej niż `val`.