

```

// W19_2.cpp : Defines the entry point for the console application.
//          Wprowadzenie w STL: algorytmy

#include "stdafx.h"
#include<iostream>
#include<vector>
#include<algorithm>

using namespace std;

void my_print(vector<int> &v);
bool UDgreater(int elem1, int elem2);

int _tmain(int argc, _TCHAR* argv[])
{
    vector<int> vec;
    vector<int>::iterator itvec;
    vector<int>::allocator_type all_typ;

    //wypelniamy wektor
    size_t it;
    int val;
    for(it=0; it<20; it++)
    {
        val = rand()%5;
        vec.push_back(val);
    }

    //wyprowadzamy na monitor
    cout << "pierwotna tablica vec\n";
    my_print(vec);

    //odnalezc wszystkie wchodzeni liczby 1
    cout << "odnalezc wszystkie wchodzeni liczby 1" << endl;
    itvec = vec.begin();
    while(itvec != vec.end())
    {
        itvec = find(itvec, vec.end(), 1);
        if(itvec != vec.end())
        {
            //powodzenie

```

```

        size_t dist = itvec-vec.begin();
        cout << "vec[" << dist << "] = 1" << endl;
        itvec++;
    }
}

//sortujemy w porzadku rosnacym
//template<class _RanIt> inline void sort(_RanIt _First, _RanIt _Last);
sort(vec.begin(), vec.end());
//wyprowadzamy na monitor
cout << "tablica vec po sortowaniu\n";
my_print(vec);

//usuwamy dublowanie danych
//template<class ForwardIterator>
//ForwardIterator unique(ForwardIterator _First, ForwardIterator _Last);
itvec = unique(vec.begin(), vec.end());
vec.erase(itvec, vec.end());
cout << "tablica vec po usunieciu dublojacych elementow\n";
my_print(vec);

//sortujemy w porzadku malejacym
sort(vec.begin(), vec.end(), UDgreater);
cout << "tablica vec po sortowaniu w porzadku malejacym\n";
my_print(vec);

vec.clear();
system("pause");
return 0;
}

void my_print(vector<int> &v)
/*=====
Uwaga! przekazanie obiektu typu vector powoduje kopiowanie tablicy
Dla tego przekazujemy referencje
=====*/
{
    size_t it;
    for(it=0; it<v.size(); it++)
    {
        cout << "it = " << it << "\t" << v[it] << endl;
    }
}

```

```
}
```

```
bool UDgreater(int elem1, int elem2)  
{  
    return elem1 > elem2;  
}
```