

Labb 3 STL

Uppgift: Använd STL (Standard Template Library) för att implementera några algoritmer genom att använda de färdiga algoritmerna och containerna i STL.

Labben kommer att återanvändas för att testa era egna implementationer av containers senare i kursen.

Genomgående så använder vi klassen **C** som element att sortera den är definierad som:

```
template <int N> struct C {  
    float value; //det är denna som används  
    int a[N];    //bara en placeholder för att det ska ta tid att  
kopiera ett C objekt.  
}
```

Observera att man med iterators inte kan ta bort ett element i en container utan att t.ex. "remove" sorterar om containern så att det som ska bort kommer sist.

Uppgift 1:

Gör ett program som:

1. Skapar en container med slumpvisa tal (C-objekt) eller bestämda tal med slumpvis ordning.
2. Skriver ut containern.
3. Tar bort alla jämna tal med hjälp av STL (använd "remove_if" och "erase").
4. Skriver ut containern.

Uppgift 2:

Gör en function

```
template <class ForwardIterator>  
void ForwardSort(ForwardIterator begin, ForwardIterator end);
```

Den ska bara använda forward iterator funktionerna (dvs. man kan göra *it, ++it, it1!=it2 och inte så mycket mera) och sortera containerna den används för.

Uppgift 3a:

Gör ett program som:

1. Skapar en container med slumpvisa tal (C-objekt) eller bestämda tal med slumpvis ordning.
2. Skriver ut containern.
3. Sorterar den baklänges mha. reverse_iterator.
4. Skriver ut containern.

Uppgift 3b:

Gör ett program som:

1. Skapar en container med slumpvisa tal (C-objekt) eller bestämda tal med slumpvis ordning.
2. Skriver ut containern.
3. Sorterar den baklänges genom att skicka med en lambda funktion som parameter.
4. Skriver ut containern.

Uppgift 4:

Gör ett program som:

1. Skapar en container med pekare till slumpvisa tal (C-objekt) eller bestämda tal med slumpvis ordning.
2. Skriver ut containern – det är C objekten som ska skrivas ut.
3. Sorterar den med `std::sort` genom att skicka med en lambda funktion som parameter – det är C objekten som ska sorteras.
4. Skriver ut containern – det är C objekten som ska skrivas ut.

Några Tips:

Random numbers: C++ bibliotek är krångligt, det gamla C varianten enkel:

```
#include <cstdlib>
```

```
int r = rand();
```

och nu är r ett random heltal.

Fylla en container med något:

```
iota(v.begin(), v.end(), 101);
```

fyller v med 101, 102, 103 etc.

```
random_shuffle(v.begin(), v.end());
```

“blandar” v

```
for (int i = 0; i < N; ++i){  
    list.push_front(C(rand()));  
}
```

Lägger till N random tal i “list”

Lambdafunktioner:

Funktionen: `[](int x){return i%10==7; }` ger true för alla tal som har 7 som sista siffra.