

객체지향프로그래밍 LAB #13

<기초문제>

1.

```
#include<iostream>
#include<vector>
using namespace std;

template<typename P, typename Q>
bool less_than(P a, Q b) { return a < b; }

template<class T>
T sum(const vector<T>& v) {
    double s = 0;
    for (int i = 0; i < v.size(); i++)
        s += v[i];
    return s;
}

int main() {
    cout << less_than(2, 3) << endl;
    cout << less_than(2.1, 2.9) << endl;
    cout << less_than(2, 2.5) << endl;

    vector<int> v1{ 1, 2, 3, 4 };
    vector<double> v2{ 10.1, 20.2, 30.3, 40.4 };
    cout << sum(v1) << endl;
    cout << sum(v2) << endl;

    return 0;
}
```



2.

```
#include<iostream>
using namespace std;

template<class T>
```

```

class Point {
private:
    T x;
    T y;
public:
    Point(T _x, T _y);
    void setXY(T _x, T _y);
    T getX();
    T getY();
    void print();
};

template<class T>
Point<T>::Point(T _x, T _y) : x(_x), y(_y) {}

template<class T>
void Point<T>::setXY(T _x, T _y) {
    x = _x;
    y = _y;
}

template<class T>
T Point<T>::getX() {
    return x;
}

template<class T>
T Point<T>::getY() {
    return y;
}

template<class T>
void Point<T>::print() {
    cout << x << ", " << y << endl;
}

int main() {
    Point<int> pt(1, 2);
    Point<double> pt2(1.1, 2.2);
    pt.print();
    pt2.print();
}

```



3.

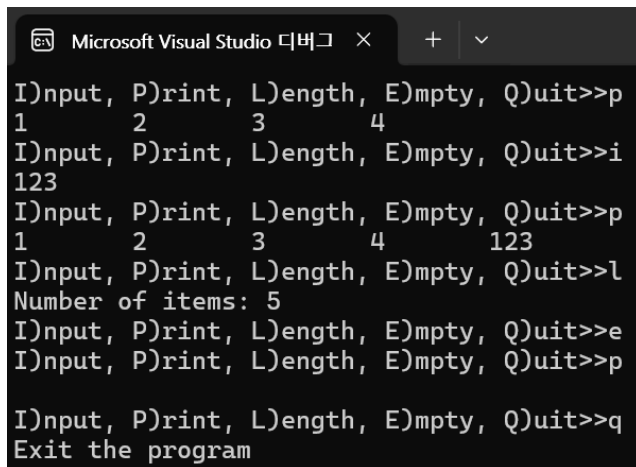
```
#include <iostream>
#include <vector> // 빠른 search, 느린 pop/push
#include <list>   // 느린 search, 빠른 pop/push
using namespace std;

int main() {
    list<int> myList{ 1, 2, 3, 4 };
    char command;
    int inputVal;
    bool finished = false;
    while (!finished) {
        //command를 입력받음
        cout << "I)input, P)rint, L)ength, E)mpty, Q)uit>>";
        cin >> command;

        //command에 따라 기능 수행
        switch (command) {
            case 'I':
            case 'i':
                cin >> inputVal;
                myList.push_back(inputVal); // push_back 구현
                break;
            case 'P':
            case 'p':
                for (const auto& elem : myList) {
                    cout << elem << " ";
                }
                cout << endl;
                break;
            case 'L':
            case 'l':
                cout << "Number of items: " << myList.size() << endl;
                break;
            case 'E':
            case 'e':
                myList.clear();
                break;
            case 'Q':
            case 'q':
                finished = true;
                cout << "Exit the program" << endl;
                break;
            default:
                cout << "Wrong command" << endl;
                break;
        }
    }

    return 0;
}
```

}



The screenshot shows the Microsoft Visual Studio Debug Console with the following text:

```
Microsoft Visual Studio 디버그 × + v
I)npur, P)rint, L)ength, E)mpty, Q)uit>>p
1      2      3      4
I)npur, P)rint, L)ength, E)mpty, Q)uit>>i
123
I)npur, P)rint, L)ength, E)mpty, Q)uit>>p
1      2      3      4      123
I)npur, P)rint, L)ength, E)mpty, Q)uit>>l
Number of items: 5
I)npur, P)rint, L)ength, E)mpty, Q)uit>>e
I)npur, P)rint, L)ength, E)mpty, Q)uit>>p

I)npur, P)rint, L)ength, E)mpty, Q)uit>>q
Exit the program
```

4.

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main() {
    int ary[] = { 1, 2, 3, 4 };
    int* pBegin, *pEnd;
    pBegin = ary;
    pEnd = ary + 4;
    for (int* plter = pBegin; plter < pEnd; plter++)
        cout << *plter << "Wt";
    cout << endl;

    //auto, begin(), end()
    vector<int> v{ 10, 20, 30, 40 };
    auto iter_begin = begin(v);
    auto iter_end = end(v);
    for (auto iter = iter_begin; iter < iter_end; iter++)
        cout << *iter << "Wt";
    cout << endl;

    return 0;
}
```



The screenshot shows the Microsoft Visual Studio Debug Console with the following text:

```
Microsoft Visual Studio 디버그 ×
1      2      3      4
10     20     30     40
```

5.

```
#include <iostream>
#include <vector> // 빠른 search, 느린 pop/push
#include <list>   // 느린 search, 빠른 pop/push
using namespace std;

template<typename Iter>
void print(const Iter& iter_begin, const Iter& iter_end) {
    for (Iter iter = iter_begin; iter != iter_end; iter++)
        cout << *iter << "Wt";
    cout << endl;
}

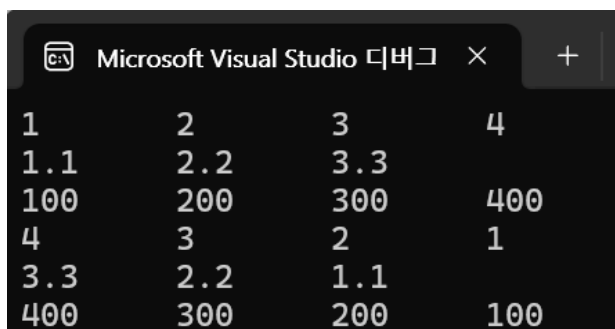
template<typename Iter>
void print_reverse(const Iter& iter_begin, const Iter& iter_end) {
    Iter iter = iter_end;
    --iter;
    for (Iter i = iter; i != iter_begin; --i)
        cout << *i << "Wt";
    cout << *iter_begin << "Wt";
    cout << endl;
}

int main() {
    vector<int> v{ 1, 2, 3, 4 };
    list<double> l{ 1.1, 2.2, 3.3 };
    int ary[] = { 100, 200, 300, 400 };

    print(begin(v), end(v));
    print(begin(l), end(l));
    print(begin(ary), end(ary));

    print_reverse(begin(v), end(v));
    print_reverse(begin(l), end(l));
    print_reverse(begin(ary), end(ary));

    return 0;
}
```



1	2	3	4
1.1	2.2	3.3	
100	200	300	400
4	3	2	1
3.3	2.2	1.1	
400	300	200	100

6.

```
#include <iostream>
using namespace std;

int sum(int x, int y) { return x + y; }
int mult(int x, int y) { return x * y; }
int evaluate(int(*f)(int, int), int x, int y) {
    return f(x, y);
}

int main() {
    cout << evaluate(&sum, 2, 3) << endl;
    cout << evaluate(&mult, 2, 3) << endl;

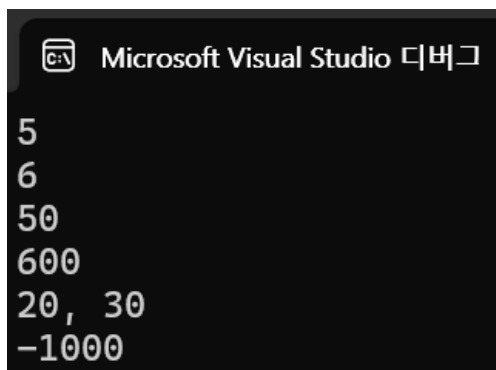
    // lambda 함수: [](입력변수)->리턴타입 {본문}
    // sum(): [](int x, int y)->int { return x + y; }
    cout << evaluate([](int x, int y)->int { return x + y; }, 20, 30) << endl;

    // simplified lambda 함수 표현: [](입력변수) {본문}
    // mult(): [](int x, int y) { return x * y; }
    cout << evaluate([](int x, int y) { return x * y; }, 20, 30) << endl;

    //생성과 호출을 동시에: 람다함수(입력값)
    [](int x, int y) { cout << x << ", " << y << endl; }(20, 30);

    auto f = [](int x, int y) { return x - y; };
    cout << f(1000, 2000) << endl;

    return 0;
}
```



7.

```
#include <iostream>
#include <functional> // function object
using namespace std;
```

```

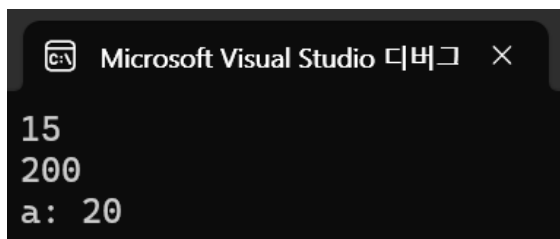
int evaluate2(function<int(int, int)> f, int x, int y) {
    return f(x, y);
}

int main() {
    int a = 10, b = 20;

    //[a]: 변수 a를 call by value로 lambda함수에 전달
    cout << evaluate2([a](int x, int y) { return a + x + y; }, 2, 3) << endl;

    //[&]: 모든 외부 변수를 call by ref.로 전달
    a = 20;
    cout << [&](int x) { return a * x; }(10) << endl;
    cout << "a: " << a << endl;
    return 0;
}

```



8.

```

#include <iostream>
#include <functional>
#include <algorithm> // for_each, copy, transform
#include <vector>
using namespace std;

int main() {
    vector<int> v1 = { 1,2,3,4 };
    for (int& elem : v1) {
        cout << elem << 'Wt';
    }
    cout << endl;

    // for_each(시작위치(iter), 끝위치(iter), 람다함수)
    for_each(begin(v1), end(v1), [](int x) {cout << x << 'Wt'; });
    cout << endl;

    for_each(begin(v1), end(v1), [](int &x) { x++; });
    for_each(begin(v1), end(v1), [](int x) {cout << x << 'Wt'; });
    cout << endl;

    int a = 10;
}

```

```

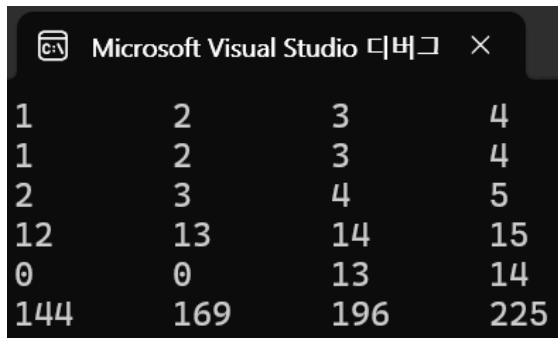
for_each(begin(v1), end(v1), [a](int &x) {x += a; });
for_each(begin(v1), end(v1), [](int x) {cout << x << 'Wt'; });
cout << endl;

vector<int> v2(v1.size());
copy(begin(v1)+1, end(v1) - 1, begin(v2)+2);
for_each(begin(v2), end(v2), [](int x) {cout << x << 'Wt'; });
cout << endl;

// transform: cont1의 element를 변형한다음(람다함수) cont2에 복사
// transform(src시작위치, src끝위치, dst시작위치, 람다함수)
transform(begin(v1), end(v1), begin(v2), [](int &x) { return x * x; });
for_each(begin(v2), end(v2), [](int x) {cout << x << 'Wt'; });
cout << endl;

return 0;
}

```



The screenshot shows the '디버그' (Debug) window in Microsoft Visual Studio. It displays a table of values, likely from a memory dump or a list of integers. The table has 4 columns and 6 rows of data.

1	2	3	4
1	2	3	4
2	3	4	5
12	13	14	15
0	0	13	14
144	169	196	225

<응용문제>

1.

```

#include <iostream>
#include <algorithm>
using namespace std;

template <typename T>
class CList
{
public:
    CList() { m_Length = 0; }
    ~CList() {}

    bool IsEmpty() {
        if (m_Length == 0)
            return 1;
        else
            return 0;
    } // list가 비어 있으면 1, 아니면 0
}

```



```

bool IsFull() {
    if (m_Length == 5)
        return 1;
    else
        return 0;
} // list가 꽉 차 있으면 1, 아니면 0

void Add(T data) {
    if (IsFull()) {
        cout << "WnWtList is full." << endl;
        return;
    }
    if (Find(data)) {
        cout << " 중복된 데이터가 존재합니다." << endl;
        return;
    }
    m_Array[m_Length] = data;
    m_Length++;
    Sortarray();
} // list에 데이터 추가
void Delete(T data) {
    if (IsEmpty()) {
        cout << "WnWtList is empty." << endl;
        return;
    }
    int index = Findindex(data);
    for (int i = index; i < m_Length - 1; i++) {
        m_Array[i] = m_Array[i + 1];
    }
    m_Length--;
} // list에 데이터 삭제
void Print() {
    cout << endl;
    cout << "Wt※ Current ListWn" << "Wt";
    for (int i = 0; i < m_Length; i++) {
        cout << m_Array[i] << " ";
    }
    cout << endl;
} // list에 데이터 출력
void Sortarray() {
    sort(m_Array, m_Array + m_Length);
}

bool Find(T data) {
    for (int i = 0; i < m_Length; i++) {
        if (m_Array[i] == data)
            return true;
    }
    return false;
}

int Findindex(T data) {
    for (int i = 0; i < m_Length; i++) {

```

```

        if (m_Array[i] == data)
            return i;
    }
    return false;
}

private:
    T m_Array[5]; // 데이터를 저장할 공간
    int m_Length; // list에 있는 데이터 수
};

int command()
{
    int num;

    cout << "WnWt---- menu ----" << endl;
    cout << "Wt1. 리스트 추가" << endl;
    cout << "Wt2. 리스트 삭제" << endl;
    cout << "Wt3. 리스트 출력" << endl;
    cout << "Wt4. 프로그램 종료" << endl;
    cout << "WnWt입력 --> ";
    cin >> num;
    return num;
}

int main()
{
    CList<int> list; // type형으로 list 선언
    int input; // list에 입력 할 데이터
    int com; // 선택한 기능
    while (1)
    {
        com = command(); // 기능을 선택
        switch (com)
        {
            case 1: // 추가
                cout << "Wn추가할 데이터 : ";
                cin >> input;
                list.Add(input);
                break;
            case 2: // 삭제
                cout << "Wn삭제할 데이터 : ";
                cin >> input;
                list.Delete(input);
                break;
            case 3: // 출력
                list.Print();
                break;
            case 4: // 프로그램 종료
                cout << "WnWt프로그램을 종료합니다Wn";
                return 0;
                break;
            default:
                break;
        }
    }
}

```

```

    }
    } return 0;
}

```

```

---- menu ----
1. 리스트 추가
2. 리스트 삭제
3. 리스트 출력
4. 프로그램 종료

입력 --> 1

추가할 데이터 : 7

```

```

---- menu ----
1. 리스트 추가
2. 리스트 삭제
3. 리스트 출력
4. 프로그램 종료

입력 --> 4

프로그램을 종료합니다

```

```

---- menu ----
1. 리스트 추가
2. 리스트 삭제
3. 리스트 출력
4. 프로그램 종료

입력 --> 1

추가할 데이터 : 9

List is full.

```

```

---- menu ----
1. 리스트 추가
2. 리스트 삭제
3. 리스트 출력
4. 프로그램 종료

입력 --> 2

삭제할 데이터 : 7

```

```

---- menu ----
1. 리스트 추가
2. 리스트 삭제
3. 리스트 출력
4. 프로그램 종료

입력 --> 2

삭제할 데이터 : 1

List is empty.

```

```

---- menu ----
1. 리스트 추가
2. 리스트 삭제
3. 리스트 출력
4. 프로그램 종료

입력 --> 1

추가할 데이터 : 7
중복된 데이터가 존재합니다.

```

```

---- menu ----
1. 리스트 추가
2. 리스트 삭제
3. 리스트 출력
4. 프로그램 종료

입력 --> 3

※ Current List
1 2 3 4 5

```

2.

```

#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    srand((unsigned int)time(0));
    vector<int> vector1(10);
    vector<int> vector2(10);
    for (int& e : vector1) {
        e = rand() % 11;
    }
}

```

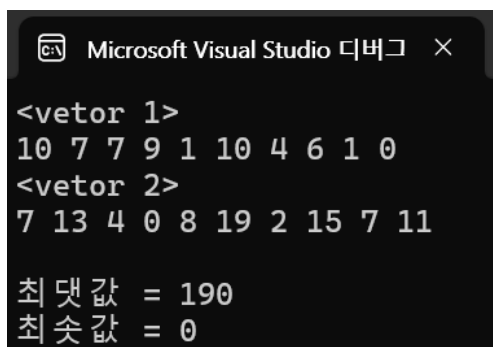
```

    }
    for (int& e : vector2) {
        e = rand() % 21;
    }
    int max_ = INT_MIN;
    int min_ = INT_MAX;

    for (int v2 : vector2) {
        for (int v1 : vector1) {
            int pr = v1 * v2;
            if (pr > max_) {
                max_ = pr;
            }
            if (pr < min_) {
                min_ = pr;
            }
        }
    }
    cout << "<vetor 1>" << endl;
    for (int e : vector1) {
        cout << e << " ";
    }
    cout << endl;
    cout << "<vetor 2>" << endl;
    for (int e : vector2) {
        cout << e << " ";
    }
    cout << endl << endl;
    cout << "최댓값 = " << max_ << endl;
    cout << "최솟값 = " << min_ << endl;

    return 0;
}

```



```

Microsoft Visual Studio 디버그 ×
<vetor 1>
10 7 7 9 1 10 4 6 1 0
<vetor 2>
7 13 4 0 8 19 2 15 7 11

최댓값 = 190
최솟값 = 0

```

3.

```

#include <iostream>
#include <string>
using namespace std;

```

```

int main() {
    while (true) {

```

```

string a;
cout << "문자열 하나를 입력해주세요 : ";
cin >> a;
if ((a == "q") || (a == "Q")) {
    break;
}
cout << "입력하신 문자열의 역순 : ";
string b = "";
for (int i = a.length() - 1; i != -1; --i) {
    b += a[i];
}
cout << b << endl;
if ([](string c, string b) {
    if (c == b)
        return true;
    else
        return false;
})(a, b))
    cout << "이 문자는 회문입니다.";
else
    cout << "이 문자는 회문이 아닙니다.";
cout << endl << endl;
}
}

```

```

C:\Users\W이인학\OneDrive - ...
문자열 하나를 입력해주세요 : LEVEL
입력하신 문자열의 역순 : LEVEL
이 문자는 회문입니다.

문자열 하나를 입력해주세요 : HELLO
입력하신 문자열의 역순 : OLLEH
이 문자는 회문이 아닙니다.

문자열 하나를 입력해주세요 : COMPUTER
입력하신 문자열의 역순 : RETUPMOC
이 문자는 회문이 아닙니다.

문자열 하나를 입력해주세요 : ABCDCBA
입력하신 문자열의 역순 : ABCDCBA
이 문자는 회문입니다.

```

4.

```

#include <iostream>
#include <vector>
#include <iomanip>
using namespace std;
void a(int n) {
    vector<vector<int>>> b(n, vector<int>(n, 0));
    int i = 0, j = n / 2;
    for (int c = 1; c <= n * n; ++c) {
        b[i][j] = c;
    }
}

```

```

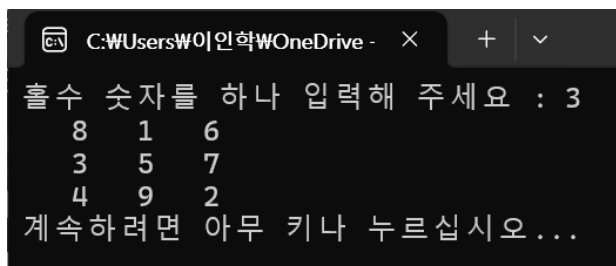
        int ni = (i - 1 + n) % n;
        int nj = (j + 1) % n;
        if (b[ni][nj] != 0) {
            i = (i + 1) % n;
        }
        else {
            i = ni;
            j = nj;
        }
    }

    for (auto row : b) {
        for (int val : row) {
            cout << setw(4) << val;
        }
        cout << endl;
    }
}

int main() {
    int n;
    cout << "홀수 숫자를 하나 입력해 주세요 : ";
    cin >> n;

    if (n % 2 == 0) {
        cout << "홀수만 입력 가능합니다.\n";
        return 0;
    }
    a(n);
    cout << "계속하려면 아무 키나 누르십시오...";
    cin.ignore();
    cin.get();
    return 0;
}

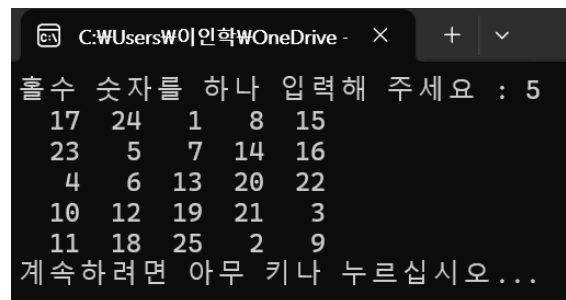
```



```

C:\Users\W0인학\OneDrive - × + v
홀수 숫자를 하나 입력해 주세요 : 3
  8  1  6
  3  5  7
  4  9  2
계속하려면 아무 키나 누르십시오...

```



```

C:\Users\W0인학\OneDrive - × + v
홀수 숫자를 하나 입력해 주세요 : 5
 17 24  1  8 15
 23  5  7 14 16
  4  6 13 20 22
 10 12 19 21  3
 11 18 25  2  9
계속하려면 아무 키나 누르십시오...

```

5.

```

#include <iostream>
using namespace std;

template <typename T>
class Queue {
private:
    T* p_list;

```

```

    int size;
    int MAX_SIZE;
public:
    Queue(int _MAX_SIZE = 1000) : MAX_SIZE(_MAX_SIZE), size(0) {
        p_list = new T[MAX_SIZE];
    }
    ~Queue() {
        delete[] p_list;
    }
    int find_index(T _item) {
        for (int i = 0; i < size; i++) {
            if (p_list[i] == _item) {
                return i;
            }
        }
        return -1;
    }

    void Enqueue(T _item) {
        if (size >= MAX_SIZE) {
            cout << "Error: out of memory" << endl;
            return;
        }
        if (find_index(_item) != -1) {
            return;
        }
        p_list[size++] = _item;
    }

    T Dequeue() {
        if (size == 0) {
            cout << "Error: No item exists in the list" << endl;
            return T();
        }
        T first_ = p_list[0];
        for (int i = 1; i < size; i++) {
            p_list[i - 1] = p_list[i];
        }
        size--;
        return first_;
    }

    void print() const {
        cout << "Items in the list : ";
        for (int i = 0; i < size; i++) {
            cout << p_list[i];
            cout << ", ";
        }
        cout << endl;
    }

    int get_size() const {
        return size;
    }

    T get_item(int _index) const {

```

```

        if ((_index < 0) || (_index >= size)) {
            cout << "Error: Index out of range" << endl;
            return T();
        }
        return p_list[_index];
    }
};

```

```

int main() {
    Queue<int> int_queue;
    Queue<float> float_queue;
    Queue<char> char_queue;

    int_queue.Enqueue(1);
    int_queue.Enqueue(2);
    int_queue.Enqueue(2);
    int_queue.Enqueue(5);

    float_queue.Enqueue(4.3);
    float_queue.Enqueue(2.5);
    float_queue.Enqueue(3.7);
    float_queue.Enqueue(3.7);

    char_queue.Enqueue('b');
    char_queue.Enqueue('b');
    char_queue.Enqueue('c');
    char_queue.Enqueue('a');

    cout << "<Before Dequeue>" << endl;
    int_queue.print();
    float_queue.print();
    char_queue.print();

    int_queue.Dequeue();
    float_queue.Dequeue();
    float_queue.Dequeue();
    char_queue.Dequeue();
    char_queue.Dequeue();
    char_queue.Dequeue();
    char_queue.Dequeue();

    cout << "<After Dequeue>" << endl;
    int_queue.print();
    float_queue.print();
    char_queue.print();

    return 0;
}

```



```
Microsoft Visual Studio 디버그 × + ▾
<Before Dequeue>
Items in the list : 1, 2, 5,
Items in the list : 4.3, 2.5, 3.7,
Items in the list : b, c, a,
Error: No item exists in the list
<After Dequeue>
Items in the list : 2, 5,
Items in the list : 3.7,
Items in the list :
```

6.

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> list{ 10, 20, 30, 40, 50 };
    int num; // 출력할 list의 수
    while (1) {
        cout << "출력 할 숫자의 수 : ";
        cin >> num;
        if (num <= 0) {
            break;
        }
        try {
            for (int i = 0; i < num; i++) {
                cout << list.at(i) << ' ';
            }
            cout << endl;
        }
        catch (exception) {
            cout << endl << "Index is out of range. Please try again.\n";
        }
    }
    cout << endl << "Program exit..." << endl;
    return 0;
}
```

Microsoft Visual Studio 디버그 × + ▾

출력 할 숫자의 수 : 2

10 20

출력 할 숫자의 수 : 9

10 20 30 40 50

Index is out of range. Please try again.

출력 할 숫자의 수 : -1

Program exit...