HW2: 新增與移除中間元素

InsertRemove.cpp

```
#include <iostream>
#include <ctime>
#include <iomanip>
#include <cstring>
#include "CBaseStack.h"
int main()
{
    int val = 0, maxSize = 20;
    std::string failToPop = "Fail to pop! Because there is no value in the array.\n";
    std::string failToRemove = "Fail to remove! Because there is no value in the array.\n";
    srand(time(NULL));
    CStack<int> baseStack;

    for (int i = 0; i < maxSize; i++)
    {
        val = rand() % 100;
        baseStack.push(val);
        std::cout << std::setw(2) << val << "  ";
        if (i % 10 == 9)
            std::cout << "\n";
    }

    for (int i = 0; i < 39; i++) // demarcation
        std::cout << (i == 38 ? "\n" : "/");
```

```
    for (int i = 0; i < maxSize; i++)
    {
        if (baseStack.pop(val))
        {
            std::cout << std::setw(2) << val << "  ";
            if (i % 10 == 9)
                std::cout << std::endl;
        }
        else
            std::cout << failToPop;
    }

    char command = '0';
    int pos = 0;
    do {
        std::cout << "Do you want to insert(i) or remove(r) any element?" << "\n";
        std::cout << "If not, press n to exit." << "\n";
        std::cin >> command;
        if (command == 'i')
        {
            std::cout << "Where do you want to insert?" << "\n";
            std::cin >> pos;
            std::cout << "What value do you want to insert?" << "\n";
            std::cin >> val;
            baseStack.insert(pos, val);
```

```
            for (int i = 0; i < 39; i++) // demarcation
                std::cout << (i == 38 ? "\n" : "/");

            maxSize++;
            for (int i = 0; i < maxSize; i++)
            {
                if ((baseStack.pop(val)) && (pos <= maxSize - 1))
                {
                    std::cout << std::setw(2) << val << "  ";
                    if ((i % 10 == 9) || (i == maxSize - 1))
                        std::cout << std::endl;
                }
                else
                    std::cout << failToPop;
            }
        }
        else if (command == 'r') // remove
        {
            std::cout << "Where do you want to remove?" << "\n";
            std::cin >> pos;
```

```
            if (baseStack.remove(pos))
            {
                for (int i = 0; i < 39; i++) // demarcation
                    std::cout << (i == 38 ? "\n" : "/");

                maxSize--;
                for (int i = 0; i < maxSize; i++)
                {
                    if ((baseStack.pop(val)) && (pos <= maxSize - 1))
                    {
                        std::cout << std::setw(2) << val << "  ";
                        if ((i % 10 == 9) || (i == maxSize - 1))
                            std::cout << std::endl;
                    }
                    else
                        std::cout << failToPop;
                }
            }
            else
                std::cout << failToRemove;
        }
    } while (command != 'n');
    return 0;
}
```

CBaseStack.h

```h
#pragma once
template<class T>
class CStack
{
private:
    const int m_StepSize;       // the unit of expand of lenght of an array
    unsigned int m_RealSize;    // the lenght of the original array
    int m_End, end;             // the address of the last element without values
    T* m_Array;                 // the array for storing values
    int pos;                    // decord the address of the array whrere user want to alter
public:
    CStack();
    ~CStack();
    bool push(T val);
    bool pop(T& val);
    bool insert(int position, T value);
    bool remove(int position);
};

template<class T>
inline CStack<T>::CStack() : m_StepSize(5), m_End(0), pos(0), end(0)
{
    m_RealSize = 5;
    m_Array = new T[m_RealSize];
}
```

```h
template<class T>
inline CStack<T>::~CStack()
{
    delete[] m_Array;
}

template<class T>
inline bool CStack<T>::push(T val)
{
    if (m_End == m_RealSize)
    {
        T* array;
        array = new T[m_RealSize + m_StepSize];
        memcpy(array, m_Array, sizeof(T) * m_RealSize);
        m_Array = array; // 離開會自動消失，把他所指的地方delete掉!
        m_Array[m_End] = val;
        m_End++;
        m_RealSize += m_StepSize;
    }
    else
    {
        m_Array[m_End] = val;
        m_End++;
    }
```

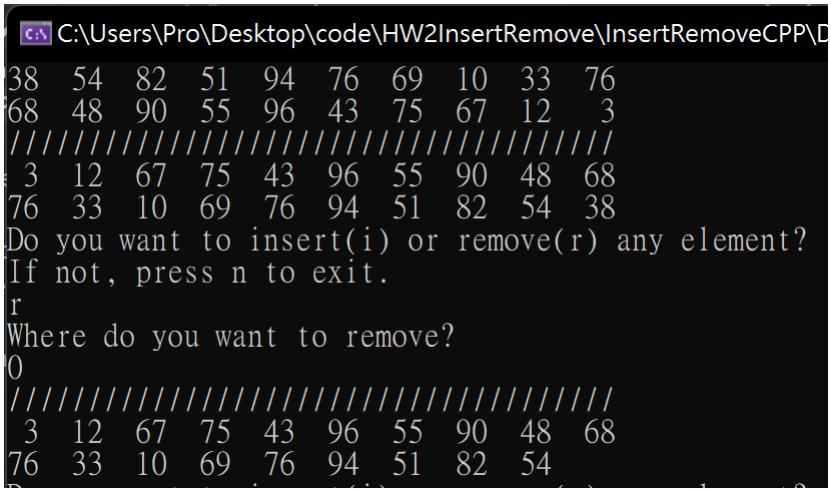
```
    end = m_End;
    return true;
}

template<class T>
inline bool CStack<T>::pop(T& val)
{
    if (m_End > 0)
    {
        m_End--;
        val = m_Array[m_End];
        return true;
    }
    else
        return false;
}
```

```
template<class T>  <T> Provide sample template arguments for IntelliSense
inline bool CStack<T>::insert(int position, T value)
{
    m_End = end;
    if (m_End == m_RealSize) // 全满
    {
        pos = position;
        T* array;
        array = new T[m_RealSize + m_StepSize];
        memcpy(array, m_Array, sizeof(T) * m_RealSize);
        m_Array = array;
        m_RealSize += m_StepSize;

        for (int i = m_End - 1; i >= pos; i--)
            m_Array[i + 1] = m_Array[i];

        m_Array[pos] = value;
        m_End++;
        end++;
    }
```

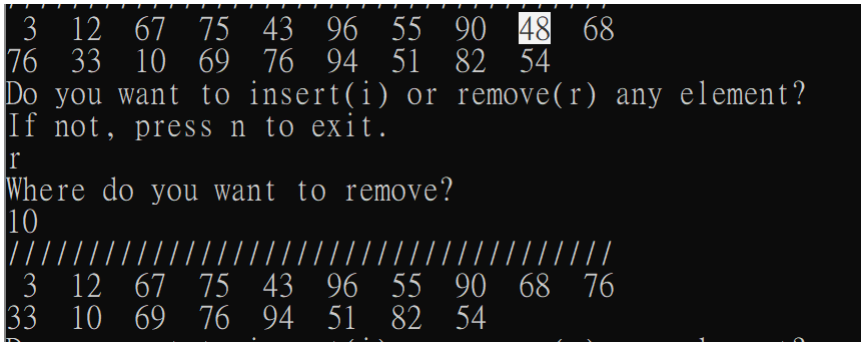
```
    else
    {
        pos = position;
        for (int i = m_End - 1; i >= pos; i--)
            m_Array[i + 1] = m_Array[i];

        m_Array[pos] = value;
        m_End++;
        end++;
    }
    return true;
}
```

```
template<class T>
inline bool CStack<T>::remove(int position)
{
    m_End = end;
    if (m_End == 0) // 全空
        return false;
    else
    {
        pos = position;
        for (int i = pos; i < m_End - 1; i++)
            m_Array[i] = m_Array[i + 1];
        m_End--;
    }
    end--;
    return true;
}
```

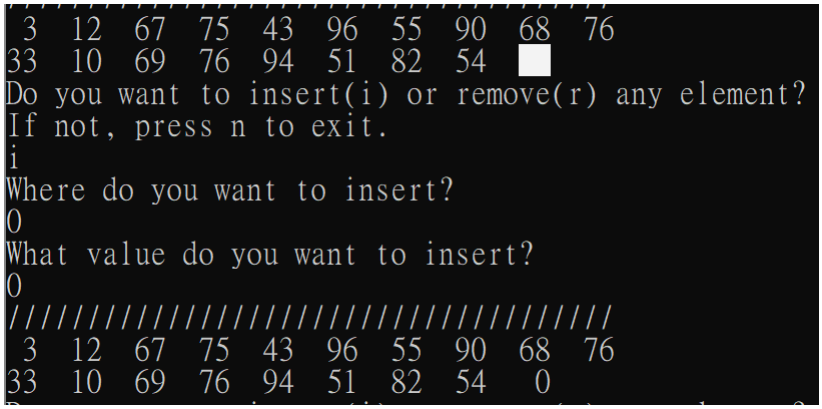
1. remove the element in index 0 and pop all elements



2. remove the element in index 10th and pop all elements



3. insert value 0 into index 0 and pop all elements



4. insert value 1 into index 10 and pop all elements

```
 3  12  67  75  43  96  55  90  68  76
33  10  69  76  94  51  82  54   0
Do you want to insert(i) or remove(r) any element?
If not, press n to exit.
i
Where do you want to insert?
10
What value do you want to insert?
1
////////////////////////////////////
 3  12  67  75  43  96  55  90  68   1
76  33  10  69  76  94  51  82  54   0
```

5. insert value 2 into index 2 and pop all elements
(Show the expansion of the array.)

```
////////////////////////////////////
 3  12  67  75  43  96  55  90  68   1
76  33  10  69  76  94  51  82  54   0
Do you want to insert(i) or remove(r) any element?
If not, press n to exit.
i
Where do you want to insert?
2
What value do you want to insert?
2
////////////////////////////////////
 3  12  67  75  43  96  55  90  68   1
76  33  10  69  76  94  51  82   2  54
 0
```

6. show “Fail to remove!” if there is no element in the array

```
////////////////////////////////////
 3  12
Do you want to insert(i) or remove(r) any element?
If not, press n to exit.
r
Where do you want to remove?
0
////////////////////////////////////
 3
Do you want to insert(i) or remove(r) any element?
If not, press n to exit.
r
Where do you want to remove?
0
////////////////////////////////////
Do you want to insert(i) or remove(r) any element?
If not, press n to exit.
r
Where do you want to remove?
0
Fail to remove! Because there is no value in the array.
```