자료구조의 기초

Lab 2. Pointer, Array

Taewhan Kim

Lab Introduction

- **Visual Studio**
  - 모랩에 설치된 **Visual Studio 2017** 사용
  - 개인 노트북에 설치된 Visual Studio 사용 가능

- **출석**
  - **출석부에 서명 + eTL에 실습 코드 업로드**로 출석 체크
  - 둘 중 하나라도 누락 시 **결석** 처리

Pointer

- Pointer
  - 변수의 **주소**를 가리키는 변수
  - 함수에 파라미터를 전달하거나,
    복잡한 데이터 타입(예 : Structure)을 사용할 때 유용

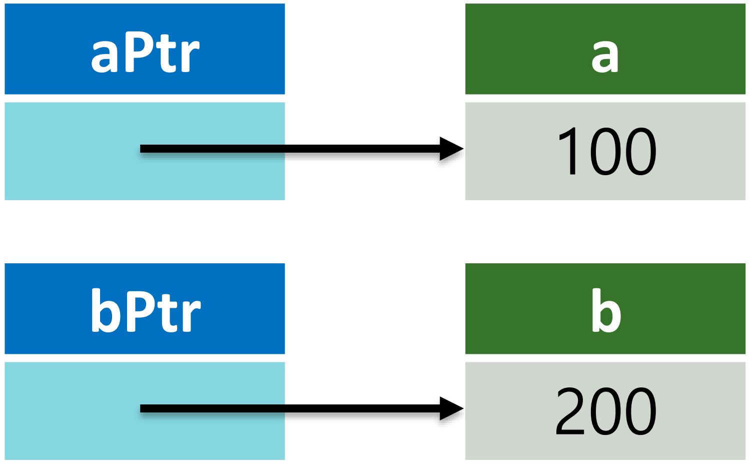
- Pointer Operator

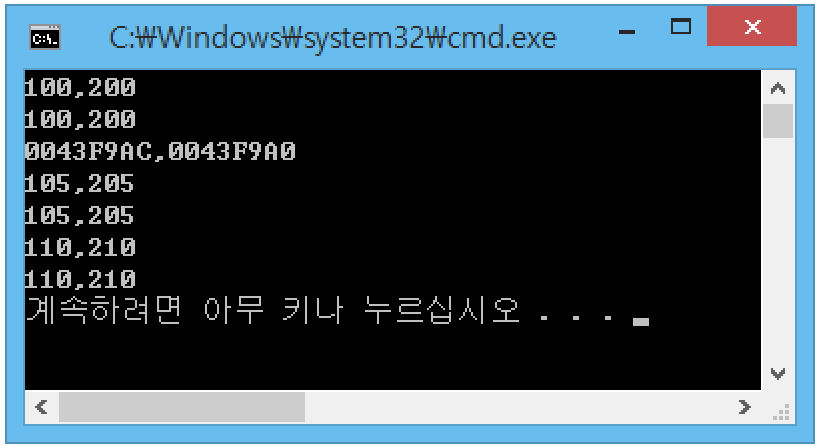
*	Pointer 변수를 선언할 때 사용 Pointer 변수가 가리키는 변수의 값에 접근할 때 사용
&	변수의 주소를 가리킬 때 사용
->	Structure의 멤버를 가리킬 때 사용

간단한 Pointer 예제

pointer.cpp

```cpp
#include <iostream>

using namespace std;

int main() {
	int a = 100;
	int b = 200;
	int* aPtr = &a;
	int* bPtr = &b;

	cout << a << "," << b << endl;
	cout << *aPtr << "," << *bPtr << endl;
	cout << aPtr << "," << bPtr << endl;

	*aPtr += 5;
	*bPtr += 5;
	cout << a << "," << b << endl;
	cout << *aPtr << "," << *bPtr << endl;

	a += 5;
	b += 5;
	cout << a << "," << b << endl;
	cout << *aPtr << "," << *bPtr << endl;

	return 0;
}
```





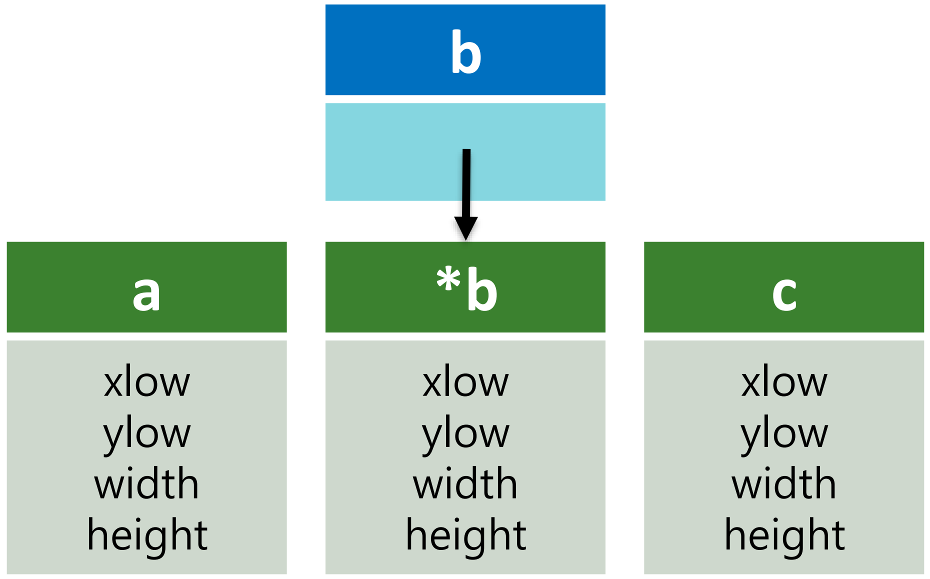
```
100,200
100,200
0043F9AC,0043F9A0
105,205
105,205
110,210
110,210
계속하려면 아무 키나 누르십시오 . . .
```

Rectangle Class 구현 – Pointer

rectangle_pointer.cpp

```
#include “rectangle.h"

int main() {
      Rectangle *b = new Rectangle();
      …
      delete b;
}
```

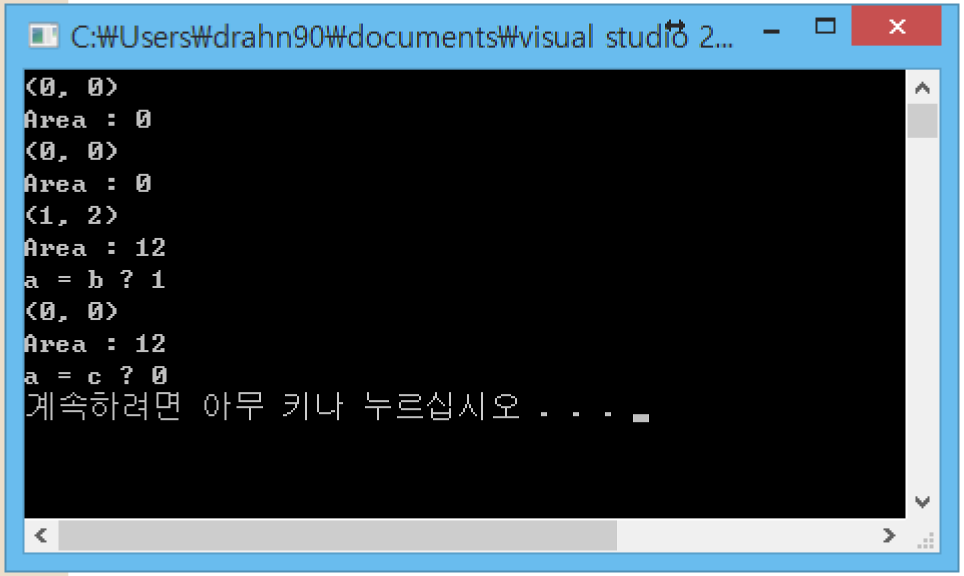
- Pointer
  - 객체의 주소를 가지고 있음
  - * 이용하여 객체에 접근
  - **new** 이용하여 메모리 할당
  - **delete** 이용하여 메모리 해제



Rectangle Class 구현 – Test

rectangle_pointer.cpp

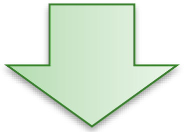
```
#include “rectangle.h"

int main() {
	Rectangle a;
	Rectangle *b = new Rectangle();
	Rectangle c(1, 2, 3, 4);

	cout << a;
	cout << *b;
	cout << c;

	cout << "a = b ? " << (a == *b) << endl;

	a.setHeight(4);
	a.setWidth(3);

	cout << a;

	cout << "a = c ? " << (a == c) << endl;

	delete b;
	return 0;
}
```



```
(0, 0)
Area : 0
(0, 0)
Area : 0
(1, 2)
Area : 12
a = b ? 1
(0, 0)
Area : 12
a = c ? 0
계속하려면 아무 키나 누르십시오 . . .
```

Array

```
int car1_weight;
int car2_weight;
int car3_weight;
int car4_weight;
int car5_weight;
int car6_weight;
        …
```



```
int car_weight[20];
```

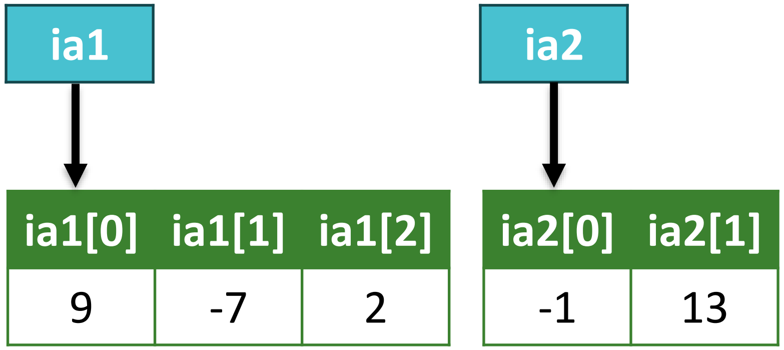
- Array
  - 동일한 자료형을 갖는 변수를 여러 개 생성할 때 하나의 Identifier 사용
  - Subscript operator([])를 이용하여 각 Element 접근
  - Pointer(*)를 이용하여도 각 Element 접근 가능
  - N개의 Element는 각각 0부터 $N-1$까지 번호 부여

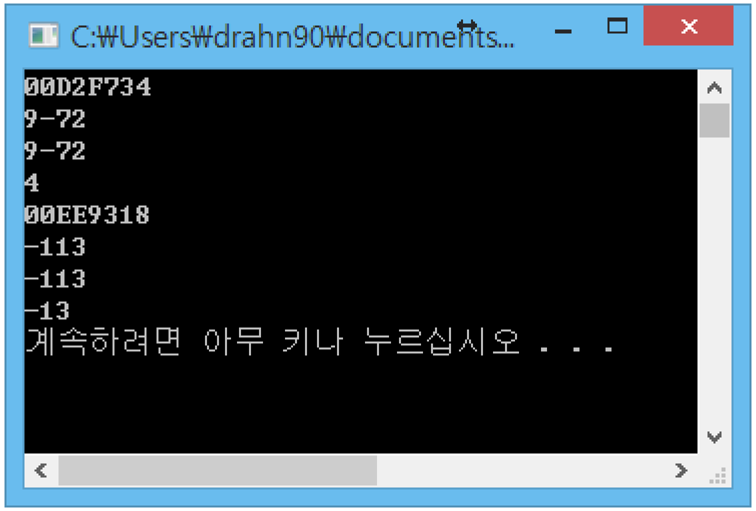
car_weight[0]	car_weight[1]	…	car_weight[19]

간단한 Array 예제

array.cpp

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

int main()
{
	int ia1[3];
	int *ia2 = new int[2];

	ia1[0] = 9; ia1[1] = -7; ia1[2] = 2;
	ia2[0] = -1; ia2[1] = 13;

	cout << ia1 << endl;
	cout << ia1[0] << ia1[1] << ia1[2] << endl;
	cout << *ia1 << *(ia1 + 1) << *(ia1 + 2) << endl;
	cout << ia1[0] + ia1[1] + ia1[2] << endl;

	cout << ia2 << endl;
	cout << ia2[0] << ia2[1] << endl;
	cout << *ia2 << *(ia2 + 1) << endl;
	cout << ia2[0] * ia2[1] << endl;

	return 0;
}
```





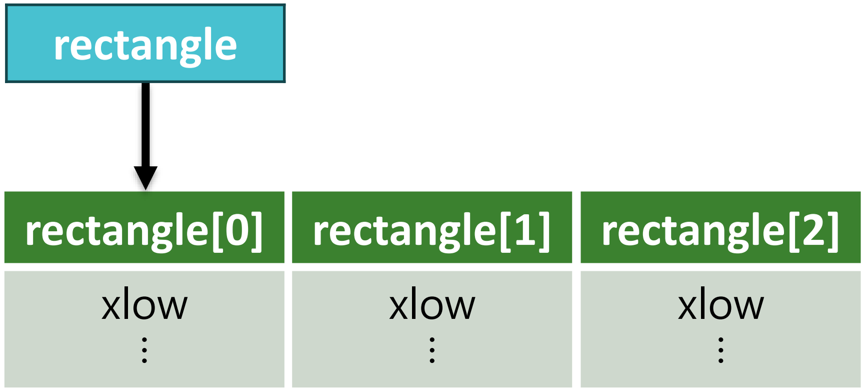
```
00D2F734
9-72
9-72
4
00EE9318
-113
-113
-13
계속하려면 아무 키나 누르십시오 . . .
```

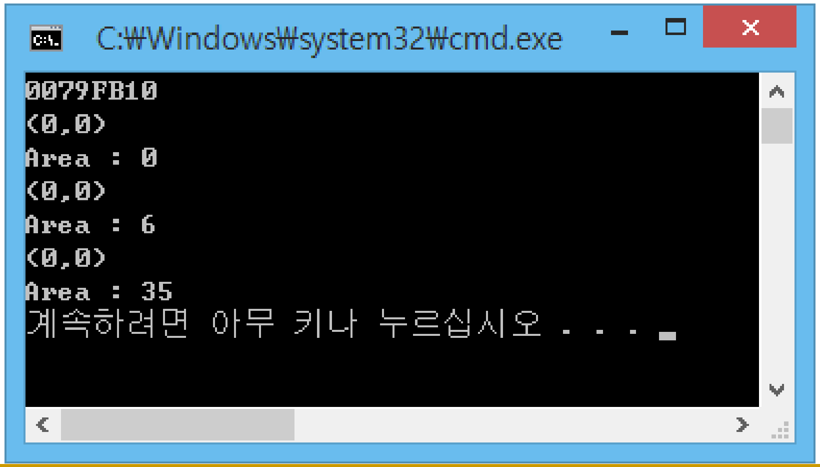
Rectangle Class 이용 – Array

rectangle_ array.cpp

```
#include "rectangle.h"
#include <iostream>
using namespace std;

int main() {
	Rectangle rectangle[3];

	rectangle[1].setWidth(2);
	rectangle[1].setHeight(3);
	(*(rectangle + 2)).setWidth(5);
	(*(rectangle + 2)).setHeight(7);

	cout << rectangle << endl;
	cout << rectangle[0];
	cout << rectangle[1];
	cout << rectangle[2];

	return 0;
}
```

- Array
  - Default constructor 이용하여
    Rectangle 객체 3개 생성





```
0079FB10
(0,0)
Area : 0
(0,0)
Area : 6
(0,0)
Area : 35
계속하려면 아무 키나 누르십시오 . . .
```

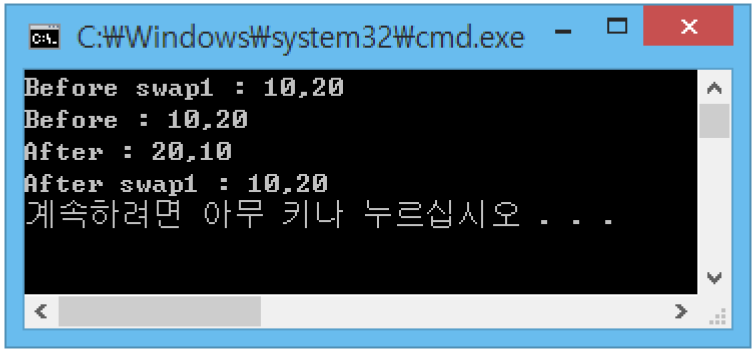
Call by Value

call_test1.cpp

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

void swap1(int a, int b)
{
	cout << "Before : ";
	cout << a << "," << b << endl;
	int temp = a; a = b; b = temp;
	cout << "After : ";
	cout << a << "," << b << endl;
}

int main() {
	int n1 = 10, n2 = 20;

	cout << "Before swap1 : ";
	cout << n1 << "," << n2 << endl;
	swap1(n1, n2);
	cout << "After swap1 : ";
	cout << n1 << "," << n2 << endl;
}
```

- Call by Value
  - 함수의 Parameter 복사하여 전달
  - 함수 호출 전후

    Parameter 값에 변화 없음



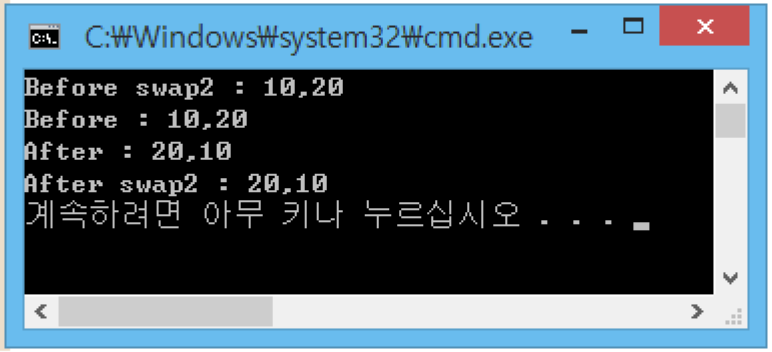
```
Before swap1 : 10,20
Before : 10,20
After : 20,10
After swap1 : 10,20
계속하려면 아무 키나 누르십시오 . . .
```

Call by Reference (1)

call_test2.cpp

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

void swap2(int *a, int *b)
{
      cout << "Before : ";
      cout << *a << "," << *b << endl;
      int temp = *a; *a = *b; *b = temp;
      cout << "After : ";
      cout << *a << "," << *b << endl;
}

int main() {
      int n1 = 10, n2 = 20;

      cout << "Before swap2 : ";
      cout << n1 << "," << n2 << endl;
      swap2(&n1, &n2);
      cout << "After swap2 : ";
      cout << n1 << "," << n2 << endl;
}
```

- Call by Reference
  - 함수의 Parameter로 주소값 전달
  - 함수 호출 전후 Parameter 값 변화



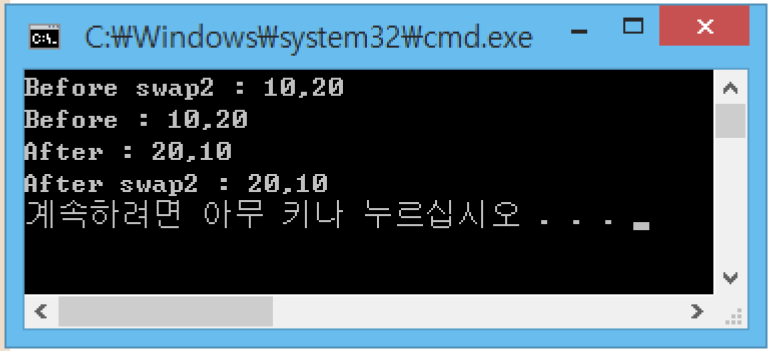
```
Before swap2 : 10,20
Before : 10,20
After : 20,10
After swap2 : 20,10
계속하려면 아무 키나 누르십시오 . . .
```

Call by Reference (2)

call_test3.cpp

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

void swap3(int &a, int &b)
{
	cout << "Before : ";
	cout << a << "," << b << endl;
	int temp = a; a = b; b = temp;
	cout << "After : ";
	cout << a << "," << b << endl;
}

int main() {
	int n1 = 10, n2 = 20;

	cout << "Before swap3 : ";
	cout << n1 << "," << n2 << endl;
	swap3(n1, n2);
	cout << "After swap3 : ";
	cout << n1 << "," << n2 << endl;
}
```

- Call by Reference
  - call_test2 결과와 동일


C:₩Windows₩system32₩cmd.exe

```
Before swap2 : 10,20
Before : 10,20
After : 20,10
After swap2 : 20,10
계속하려면 아무 키나 누르십시오 . . .
```

TODO : 큰 수 덧셈 구현

- **설명**
  - Integer 표현 범위를 벗어나는 매우 큰 자연수 A, B의 합 A+B를 Array와 Pointer를 이용하여 계산

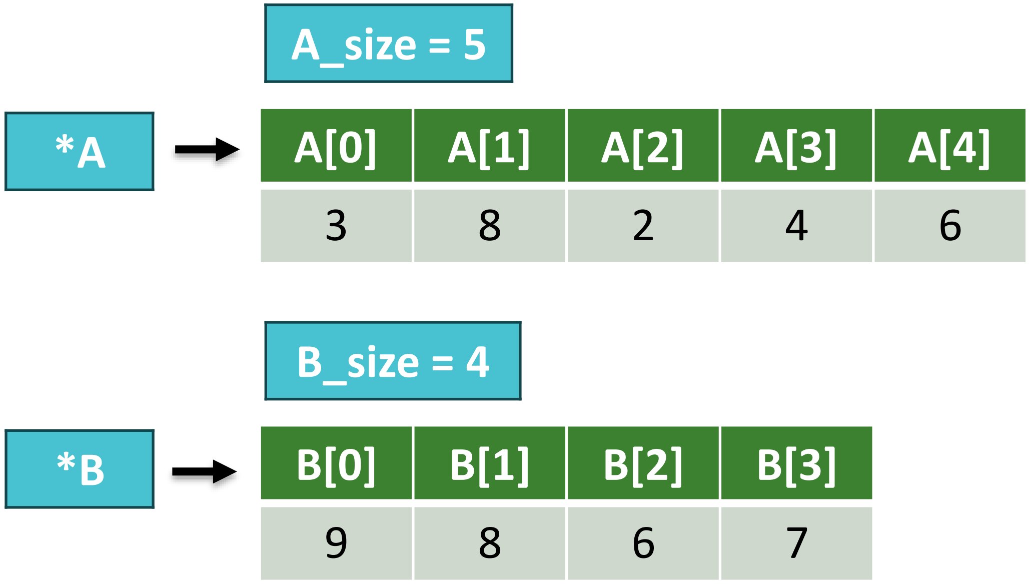
- **TODO**
  - eTL에 주어진 BigInt.cpp의 CalculateSum 함수 구현
  - CalculateSum (int *A, int A_size, int *B, int B_size)
    - *A, *B : 두 자연수 A, B가 Array 형태로 저장된 위치를 가리키는 Pointer
    - A_size, B_size : 두 자연수 A, B 각각의 자릿수
    - 두 수의 합 계산 뿐만 아니라 계산한 값의 출력도 구현해야 함

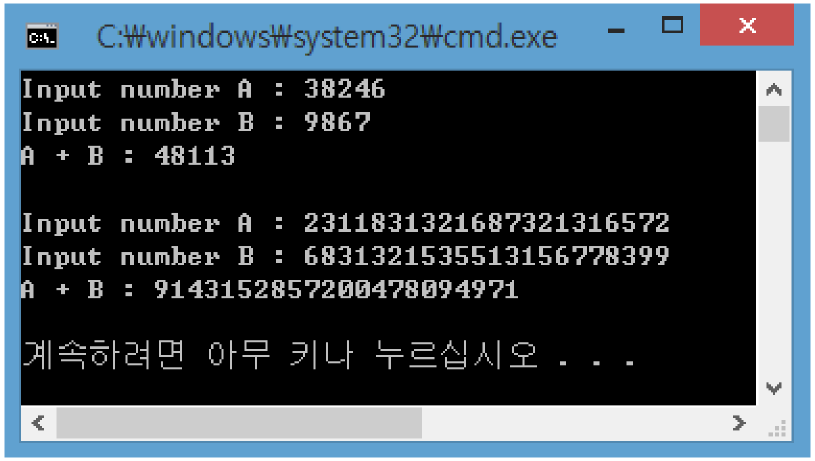
TODO : 큰 수 덧셈 구현

- **Test**

BigInt.cpp

```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {

	string As, Bs;
	int A_size, B_size;
	int *A, *B;

	cout << "Input number A : "; cin >> As;
	cout << "Input number B : "; cin >> Bs;

	A_size = As.size();
	B_size = Bs.size();

	A = String_To_Int(As);
	B = String_To_Int(Bs);

	CalculateSum(A, A_size, B, B_size);

	delete[] A; delete[] B;
}
```





```
Input number A : 38246
Input number B : 9867
A + B : 48113

Input number A : 2311831321687321316572
Input number B : 6831321535513156778399
A + B : 9143152857200478094971

계속하려면 아무 키나 누르십시오 . . .
```

Code Submission

- **코드 제출**
  - 완료한 실습 코드를 다음과 같이 압축
    - 제출할 코드 : pointer.cpp, rectangle_pointer.cpp, array.cpp, rectangle_array.cpp, call_test1.cpp, call_test2.cpp, call_test3.cpp, BigInt.cpp
    - 압축 파일명 : lab2_홍길동_2017-10000.zip
  - 오늘 (2018년 3월 14일) **오후 11시**까지 eTL에 제출
  - 제출된 코드는 따로 채점하지 않음

- **출석**
  - **출석부에 서명 + eTL에 실습 코드 업로드**로 출석 체크
  - 둘 중 하나라도 누락 시 **결석** 처리