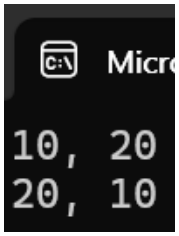
객체지향프로그래밍 LAB #08

<기초문제>_____

1.

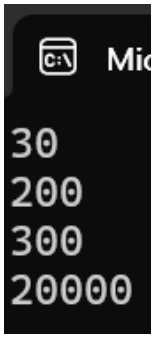
```
#include<iostream>
using namespace std;
//int &x = a; int &y = b; pass by reference
//int *x = &a; int *y = &b; pass by address
void swap(int* x, int* y) {
	int temp = *x;
	*x = *y;
	*y = temp;
}
int main() {
	int a = 10, b = 20;
	cout << a << ", " << b << endl;
	swap(&a, &b);
	cout << a << ", " << b << endl;
	return 0;
}
```



2.

```
#include<iostream>
using namespace std;
int sum(int x, int y) { return x + y; }
int mult(int x, int y) { return x * y; }
// 함수 func의 포인터와 파라미터 x, y를 받아와서 함수 func의 실행값을 리턴해주는 함수
int evaluate(int(*func)(int, int), int x, int y)
{
	return func(x, y);
}
```

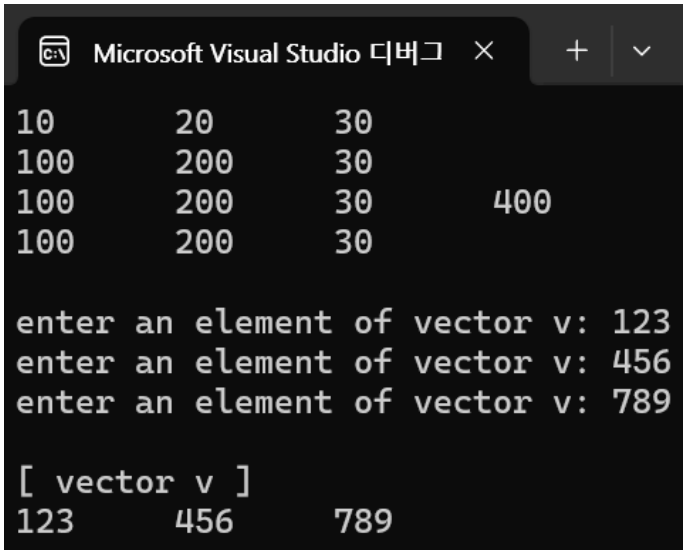
```
int main() {
	int(*func)(int, int);
	func = &sum; // sum 함수 주소값 대입
	cout << func(10, 20) << endl;
	func = &mult; // mult 함수 주소값 대입
	cout << func(10, 20) << endl;
	cout << evaluate(&sum, 100, 200) << endl;
	cout << evaluate(&mult, 100, 200) << endl;
	return 0;
}
```



3.

```
#include<iostream>
#include<vector>
using namespace std;
void print(const vector<int>& v) {
	for (unsigned i = 0; i < v.size(); i++)
		cout << v[i] << "₩t";
	cout << endl;
}
int main() {
	vector<int> vec{ 10, 20, 30 };
	print(vec);
	vec[0] = 100; // vector 인덱스 0번에 직접 대입
	vec.at(1) = 200; // vector의 at() 함수 이용
	print(vec);
	vec.push_back(400); // vector의 push_back() 함수 이용
	print(vec);
	vec.pop_back(); // vector의 pop_back() 함수 이용
```

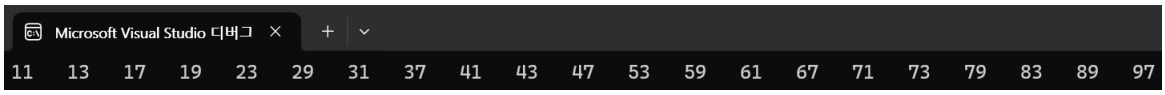
```
	print(vec);
	cout << endl;
	// 범위기반 for문을 이용한 vector 원소값 바꾸기 (슬라이드 p.26 참고)
	for (int& elem : vec) {
		cout << "enter an element of vector v: ";
		cin >> elem;
	}
	cout << endl << "[ vector v ]" << endl;
	// 범위기반 for문을 이용한 화면 출력하기 (슬라이드 p.26 참고)
	for (int elem : vec)
		cout << elem << "₩t";
	cout << endl;
	return 0;
}
```



4.

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

bool is_prime(int num) {
	if (num < 2)
		return false;
	for (int i = 2; i < num; i++) {
```

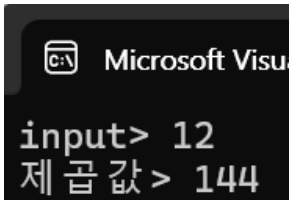
```
                if (num % i == 0) {
                        return false;
                }// num을 i로 나눈 나머지가 0이면 false 리턴
        }
        return true;
}

vector<int> primes(int lower, int upper) {
        vector<int> v;
        for (int i = lower; i <= upper; i++) {
                if (is_prime(i)) {
                        v.push_back(i);
                } // 소수이면(is_prime이 참이면) 뒤에 push
        }
        return v;
}

int main() {
        int lower = 10, upper = 100;
        vector<int> vec = primes(lower, upper); // 벡터 vec 선언 및 primes로 대입
        for (int elem : vec)
                cout << elem << setw(5);
        cout << endl;

        return 0;
}
```



<응용문제>_____

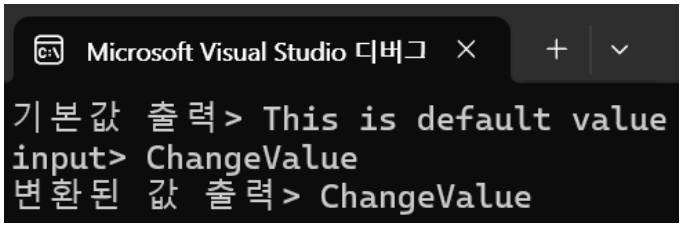
1.

```
#include<iostream>
using namespace std;
void s(int* x) {
	*x = (*x) * (*x);
}
int main() {
	int a;
	cout << "input> ";
	cin >> a;
	s(&a);
	cout << "제곱값> " << a;
}
```



2.

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

void s(string* a) {
	cout << "input> ";
	cin >> *a;
}
int main() {
	string str = "This is default value";
	cout << "기본값 출력> " << str << endl;
	s(&str);
	cout << "변환된 값 출력> " << str << endl;
	return 0;
}
```



3.

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

void f(vector<int>& a) {
	for (int c = 0; c + 1 <= int(a.size() / 2); c++) {
		int temp = a[c];
		a[c] = a[a.size() - 1 - c];
		a[a.size() - 1 - c] = temp;
	}
}
int main() {
	vector<int> v(10);
	int i = 1;
	for (int& e : v) {
		e = i;
		i++;
	}
	cout << "기본 Vector 값:" << endl;
	for (int e : v) {
		cout << ' ' << e;
	}
	f(v);
	cout << endl;
	cout << "함수 실행 후 Vector 값:" << endl;
	for (int e : v) {
		cout << ' ' << e;
	}
}
```

4.

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

int inputGrade(float grade) {
	cin >> grade;
	if ((grade - (int)grade) != 0.0 || grade < 0 || grade > 100) {
		cout << "invalid input!!!" << endl;
		exit(EXIT_FAILURE);
	}
	else
		return (int)grade;
}
int topGrade(const vector<int>& grade) {
	int maximum = grade[0];
	if (grade[1] > maximum)
		maximum = grade[1];
	if (grade[2] > maximum)
		maximum = grade[2];
	return maximum;
}
int main() {
	vector<int> grades;
	float _grade = 0;
	for (int i = 0; i < 3; i++) {
		cout << "Input " << i << "-th grade(0~100):";
		int grade = inputGrade(_grade);
		grades.push_back(grade);
	}
	int d = topGrade(grades);
```

```
        cout << "Top Grade: " << d;
                return 0;
}
```