

C언어 스터디

3.5주차

<C++ 맷보기 && 동적배열(vector) && 문자열>



CAPS

C++ 맛보기

1. C Style

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int main(void) {
    printf("Hello World!");
}
```

2. C++ Style

```
#include<iostream>
using namespace std; // 이게 뭐예요? 모르셔도 됩니다. 심프 때 배우십니다.
int main(void) {
    cout << "Hello World";
}
```

1. C Style

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int main(void) {
    int a, b;
    scanf("%d %d", &a, &b); // C언어는 주소값을 넘겨줘야 하지만..
    printf("%d", a + b);
}
```

2. C++ Style

```
#include<iostream>
using namespace std;
int main(void) {
    int a, b;
    cin >> a >> b; // 주소값을 안 넘겨줘도 됩니다! (그렇게 코딩 되어 있음)
    cout << a + b;
}
```

<https://www.acmicpc.net/problem/10950>

1. C Style

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int main(void) {
    int T;
    scanf("%d", &T);
    while (T--) {
        int a, b;
        scanf("%d %d", &a, &b);
        printf("%d\n", a + b);
    }
}
```

2. C++ Style

```
#include<iostream>
using namespace std;
int main(void) {
    int T;
    cin >> T;
    while (T--) {
        int a, b;
        cin >> a >> b;
        cout << a + b << "\n";
    }
}
```

동적배열

1. C Style

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<stdlib.h> // standard library 헤더 추가해야 함!
int main(void) {
    int* arr = (int*)malloc(sizeof(int) * 10);
    //printf("%d", arr[11]); error!
    arr = (int*)realloc(arr, sizeof(int) * 100);

    arr[11] = 100;
    printf("%d", arr[11]);
}
```

2. C++ Style

```
#include<iostream>
#include<vector> // vector 라이브러리 추가해야 함!
using namespace std;
int main(void) {
    vector<int> v(10);
    //cout << v[11]; error!
    v.resize(100);

    v[11] = 10101;
    cout << v[11];
}
```

3. C++ Style + a

```
#include<iostream>
#include<vector>
using namespace std;
vector<int> v; // {}
int main(void) {
    for (int i = 0; i < 5; i++) {
        int a;
        cin >> a;
        v.push_back(a);
    } // {1,2,4,5,6}
    cout << "\n" << v.size() << "\n"; //5
    for (int i = 0; i < v.size(); i++) cout << v[i] << " "; // 1 2 4 5 6

    v.pop_back(); // {1,2,4,5}
    v.pop_back(); // {1,2,4}
    cout << "\n" << v.size() << "\n"; //3
    for (int i = 0; i < v.size(); i++) cout << v[i] << " "; // 1 2 4

    v.resize(6); // 6
    cout << "\n" << v.size() << "\n"; // {1,2,4,0,0,0}
    for (int i = 0; i < v.size(); i++) cout << v[i] << " "; // 1 2 4 0 0 0
    cout << "\n";
    v[3] = 100; // {1,2,3,100,0,0}
    cout << v.front() << " " << v.back(); // 1 0
    cout << "\n";
    printf("%p %p", v.begin(), v.end()); // 배열의 시작과 끝의 포인터값입니다.
    // 000000AA259AFD20 000000AA259AFD60
    // 아주 엄밀히 이야기해서 포인터는 아니지만, 비슷합니다.
}
```

문자열

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
char arr[8] = "Hello";
int main(void) {
    printf("%s\n", arr);
    for (int i = 0; i < 8; i++) printf("%c", arr[i]);
}
```

index	0	1	2	3	4	5	6	7
value	H	e	l	l	o	0	0	0

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
char arr[8] = "Hello"; // C스타일의 문자열은 원소가 char인 배열입니다.
int main(void) {
    for (int i = 0; i < 8; i++) printf("%d ", (int)arr[i]);
}
```

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
char arr[100];
int main(void) {
    scanf("%s", arr);
    printf("%s", arr);
}
```

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
char arr[8] = "Hello";
int main(void) {
    arr[2] = 0;
    printf("%s", arr); // He
    // He0lo000 으로 저장되어 있다면, 왼쪽부터 읽다가 0에서 멈춥니다.
}
```

```
#include<iostream>
using namespace std;
int main(void) {
    string arr; // C++에서는 기본적으로 string 자료형을 지원합니다.
    cin >> arr;
    cout << arr;
}
```

```

#include<iostream>
using namespace std;
int main(void) {//string은 vector처럼 사용할 수 있습니다.
    string str = "Hello";           ///"Hello"
    str.push_back('K');            ///"HelloK"
    cout << str << '\n';          //HelloK
    cout << str.size() << " " << str.length() << '\n'; //5 5
    cout << str.front() << " " << str.back() << '\n'; //H K
    str.pop_back();                ///"Hello"
    str.pop_back();                ///"Hell"
    str[0] = 'A';                  ///"Aell"
    cout << str;                  //Aell
}// 추가적으로 쓰고 싶은 게 있으면 #include<string>을 해 보세요!

```

C++ 템플릿

```

#include<iostream>
using namespace std;
int main(void) {

}

```

Fast C++ 템플릿

```

#include<iostream>
using namespace std;
int main(void) {
    ios::sync_with_stdio(0);
    cin.tie(0);

}
/*
이제 백준에서 문제를 푸실 때는 해당 템플릿에 코드를 채우게 될 겁니다.
ios::sync_with_stdio(0);
cin.tie(0);
이 둘은 입출력을 빠르게 해줍니다. 자세한 설명은 여백이 부족해 생략할게요.
사실 별로 안 중요합니다.
따로 배워서 상황에 맞게 적절한 라이브러리를 잘 추가해 주세요. 파이팅입니다.
*/

```

문제)

1. 11654 아스카| 코드

<https://www.acmicpc.net/problem/11654>

<https://github.com/sungjoonyoung/BOJ/blob/main/10000/11654.cpp>

생각보다 더 많이 쓰일 테크닉입니다.

2. 11720 숫자의 합

<https://www.acmicpc.net/problem/11720>

<https://github.com/sungjoonyoung/BOJ/blob/main/10000/11720.cpp>

테크닉 자리죠?

3. 9086 문자열

<https://www.acmicpc.net/problem/9086>

<https://github.com/sungjoonyoung/BOJ/blob/main/00000/9086.cpp>

string의 함수를 잘 써봅시다.

4. 2744 대소문자 바꾸기

<https://www.acmicpc.net/problem/2744>

<https://github.com/sungjoonyoung/BOJ/blob/main/00000/2744.cpp>