C++에 대하여

C언어의 업그레이드 버전!

: C언어를 cpp로 저장하고 사용해도 문제 X

파일의 확장자를 .c 대신 .cpp로 저장!

사용하는 주 이유

: C언어보다 더 많은 라이브러리를 가지고 있음

#include <cstdio>

#include <stdio.h>와 다른 건 없지만 메모리를 덜 사용함

printf()와 scanf()를 포함하고 있음

#include <cstring>

#include <cstdio>와 마찬가지로 c언어 에서의 <string.h>와 같음

strlen(string a) : 문자열 a의 길이를 반환

strcmp(string a, string b) : a, b를 사전순으로 비교

// a가 앞에 올 경우 음수, b가 앞에 올 경우 양수 반환

strcpy(string a, string b) : a에 b를 복사

strcat(string a, string b) : a 뒤에 b를 이어 붙임

#include <cstdlib>

abs(int a) : 절댓값 a 반환

atoi(string a) : 문자열 a를 받아 int 형으로 반환

#include <cmath>

sqrt(int a) : a의 제곱근 반환

pow(int a, int b) : a의 b제곱 반환

#include <iostream>

int a;

cin >> a; // 입력 받기

cout << a << endl; // 출력 후 줄바꿈 화살표 방향 조심!

#include <algorithm>

using namespace std; // 몇 개의 헤더에서는 선언해주어야 함 -> 그냥 쓰는 습관 ㄱ

알고리즘에 관련된 라이브러리 포함

std::sort(arr, arr+N) : 배열을 N개까지 오름차순 정렬

std::min(int a, int b) : a와 b 중 작은 수 반환

std::max(int a, int b) : a와 b 중 큰 수 반환

std::은 using namespace std;를 선언하지 않으면 사용 함수 앞에 적어줘야 함

#include <string>

clear() : 문자열을 비우는 역할

string.size() : 스트링의 사이즈를 반환

string.substr(int a) : a번째 인덱스부터 끝까지의 문자열을 반환

string,substr(int a, int b) : 인덱스 a부터 b까지의 문자열을 반환(문자열의 부분)

string.c\_str() : 스트링을 %s를 이용하여 출력할 때 사용

ex) printf(“%s”, string.c\_str());

string.find(“abc”) : 문자열에서 abc를 찾는 함수

Vector에 대하여 ( #include <vector> )

배열과 비슷한 정보 저장 장소

배열처럼 vector[i]로 접근 가능

vector < 자료형 > 변수 이름; // 선언

vector.push\_back(a); // 벡터에 자료를 넣는 방법

vector.pop\_back(a) // 벡터에서 맨 뒤의 자료를 삭제 하는 명령

sort(vector.begin(), vector.end());

Pair에 대하여 ( #include <utility> )

pair < 자료형, 자료형 > 변수이름; // 선언 방법

페어로 여러 개도 선언 가능!

pair.first << 페어의 첫번째 자료형에 접근하는 방법

pair.second << 페어의 두번째 자료형에 접근하는 방법

페어는 sort할 경우 첫 번째 자료형을 우선순위로 정렬 -> sort(v.begin(), v.end())