

삼성 청년 SW 아카데미

알고리즘

<알림>

본 강의는 삼성 청년 SW아카데미의 컨텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

Day3. String

컴퓨터는 0/1 만 아는 데, 왜 String 을 알아야 하나?!

- User는 이진수로 입력하지 않는다.
- 문자열은 사람이 이해하기 쉬운 형태로 데이터를 처리하므로 개발자도 이해하기 쉽다.

작업의 효율성을 높이기 위해서도 문자열은 반드시 알아야 한다.

물론, 코딩 테스트의 관점에서도 이진수 처리보다 까다로우니
문자열 처리 문제로 허수 거르기가 용이하다

C++ 에서는 char 배열 이외에 string 을 지원한다.

- <string> 필요
- string a;

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

int main() {
    string a;
```

코드를 작성하고 빌드한다.

string의 복사 및 비교

- 매우 편하다

<https://gist.github.com/hoconoco/08ff6bfeabf440c320d5ab0d4bf2a979>

```
string a = "AAA";
string b;

//복사하기
b = a;
cout << b << '\n';

//비교하기
if (a == b) {
    cout << "같다.";
}
```

코드를 작성하고 빌드한다.

문자열 더하기

- 매우 편하다.

<https://gist.github.com/hoconoco/8872a3f50cabdb3f9297aa8560b40106>

```
string a = "AAA";  
string b = "BBB";
```

```
//문자열 더하기  
cout << a + b;
```

코드를 작성하고 빌드한다.

문자열 길이 구하기

- `.size()` : STL 용 API
- `.length()` : string 용 API
- 매우 편하다.

```
string a = "AAA";
string b = "BBB";

cout << a << " " << a.length() << " " << a.size();
```

<https://gist.github.com/hoconoco/8715004147e4a1ea43a51271754dd28b>

문자열 찾기

int find("찾을 문자열", 검색할 위치)

- 찾을 문자열을 원하는 위치에서 검색할 수 있음, 검색 위치는 default = 0
- return : 문자열 중 찾을 문자열의 위치 반환, 없으면 -1
 - string::npos 반환

<https://gist.github.com/hoconoco/81c8082d8b180b22f7edefe02333ec10>

```
string a = "ABCKFCABC";  
  
//있으면 index 반환  
int ret = a.find("KFC");  
cout << ret << '\n';  
  
//없으면 -1, string::npos 반환  
//int 최대 값 -> -1  
ret = a.find("HIFAKER");  
cout << ret << '\n';  
  
//검색 위치 설정 가능, 0이 default  
ret = a.find("ABC", 0);  
cout << ret << '\n';  
  
//검색 위치 설정 가능  
ret = a.find("ABC", 3);  
cout << ret << '\n';
```

오른쪽 문자열에서 BB 찾기

- 결과 OXOX

```
string vect[4] = {  
    "ABBT",  
    "BTBT",  
    "BBBBT",  
    "KFC"  
};
```

문자열 역순 출력하기

```
TBBA
&TBTB
#TBBBB
CFK

C:\Users\mincod
ai すと なと こ
```

```
string vect[4] = {
    "ABBT",
    "BTBT&",
    "BBBT#",
    "KFC"
};
```

문자열 잘라내기

string .substr(index, cnt)

- return : 해당 index 부터 cnt(없으면 끝)까지 반환

```
string a = "ASDKODWQKDMQWD[1234]KALSMQLKASMD";
//a에서 1234만 잘라내기

//1. 찾기
int Index = a.find('[');
//2. 잘라내기
string ans = a.substr(Index + 1, 4);
cout << ans;
```

<https://gist.github.com/hoconoco/b0a666c571938da4186019475cb068a3>

문자열 a 가 있다. (A~Z 까지 알파벳 순서대로)

숫자 2개 입력 받아 a~b index 까지 출력하자

```
string a = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

2 5
CDEF

오른쪽 배열에서 숫자만 출력하기

```
string vect[4] = {  
    "ABCQ",  
    "B[4]AS",  
    "C[34]C",  
    "BT[15]"  
};  
//4 34 15 출력하기
```

문자열을 정수형으로 변환

stoi(문자열)

정수형을 문자열로 변환

to_string(정수)

```
string str = "123";
//문자열을 정수로
cout << stoi(str) + 100 << '\n';

int a = 123;
//정수를 문자열로
cout << to_string(a) + "100";
```

<https://gist.github.com/hoconoco/befecb408dcdfce502232bf82d583429>

문자열 a에서 숫자만 더해서 출력하자

- $45 + 123213 = 123258$

```
string a = "AB[45]AB[123213];"
```

123258

문자열 str이 있다. ABC 개수를 출력한다

- 3

```
string str = "AB[ABC]ASDF[ABC]SDF[ABC]";
```

3

문자열 vect 가 있다.

GOLD 개수 출력하기

- 5

```
string vect[3] = {  
    "BTSGOLD",  
    "GOLDGOLD",  
    "GOLDBARGOLD"  
};
```

5

문자열 str이 있다

숫자들의 합을 계산해서 출력

- $1231+1231+434+77+123 = 3096$

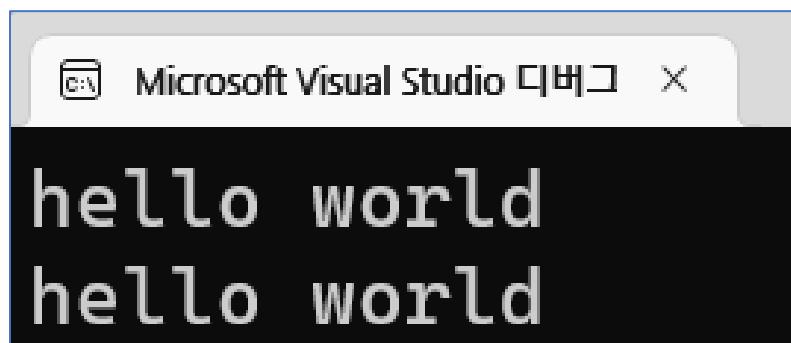
```
string str = "1231-1231-434-77-123";
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	2	3	1	-	1	2	3	1	-	4	3	4	-	7	7	-	1	2	3

C++에서 띄워쓰기 있는 문장 입력 받기

`getline(cin, string);`

- 파일 명이 .cpp 여야 한다.



```
string a;  
  
getline(cin, a);  
cout << a;
```

<https://gist.github.com/hoconoco/795c00d502aafbadf49b50e7177bc48d>

C에서 띄워쓰기 있는 문장 입력 받기

```
scanf("%[^\\n]", a);  
fgets(a, sizeof(a), stdin);
```

- 파일 명이 .c 여야 한다.

```
hi faker  
hi faker
```

```
char a[100];  
scanf("%[^\\n]", a);  
printf("%s\\n", a);
```

```
//fgets(a, sizeof(a), stdin);  
//printf("%s", a);
```

내일
방송에서
만나요!

삼성 청년 SW 아카데미