

문자열 함수 구현 (각 함수당 수행시간 1초)

기존 standard library에 있는 다음의 6개의 스트링 함수의 사용법을 익히고 구현하도록 한다 (기존의 함수와 똑같이 작동해야 함). 또한, 추가로 2개의 스트링 함수를 구현한다.

```
size_t strlen(const char *s);
```

리턴값 : 문자열의 길이

```
char *strcat(char *dest, const char *src);
```

리턴값 : dest의 포인터가 리턴된다.

```
char *strrev(char *s); (리눅스 및 유닉스 계열에서 지원 하지 않음)
```

리턴값 : 역순으로 바꾼 문자열 s의 번지

```
int strcmp(const char *s1, const char *s2);
```

리턴값 : s1 < s2 인 경우 음수를 리턴

s1 == s2 인 경우 0을 리턴

s1 > s2 인 경우 양수를 리턴

```
char *strcpy(char *dest, const char *src);
```

리턴값 : dest의 포인터를 리턴한다.

```
char *strstr(const char *s1, const char *s2);
```

리턴값 : s1에서 부분문자열이 나오는 첫 번째 포인터를 리턴한다, 부분 문자열 s2가 s1내에서 발견 되지 않으면 NULL을 리턴한다

(reference : <http://winapi.co.kr>)

```
size_t strreplace(char *s, const char *s1, const char *s2);
```

함수 설명 : 문자열 s에서 s1을 찾아 s2로 치환한다.

리턴값 : 치환 된 문자열의 개수를 리턴한다.

```
char *trim(char *s);
```

함수 설명 : 문자열의 앞뒤 space 문자 (\t, \n, ' ' 등)를 잘라낸다.

(isspace() 함수를 사용하여 space 문자를 판별한다)

리턴값 : 잘라낸 문자열의 첫 번째 포인터를 리턴한다.

1. main.c 파일은 주어지고 string.c 파일만 수정하도록 한다. (gcc -O2 main.c string.c)
(제출은 string.c만 하며, 주어진 main.c와 컴파일을 한다. 컴파일 되지 않으면 0점, main.c는 단지 테스트를 위해서만 제공 된 것이며 직접 만들어도 무관하다.)
2. 각 함수의 소스는 bracket ('[', ']')을 절대 사용하지 않는다. bracket 사용 시 50% 감점.
3. #include <string.h>를 사용하면 0점 처리한다. (string 라이브러리 사용 불가)