

과제 1번

일련의 부동 소수 값들의 평균을 구하는 `avg()`라는 함수를 작성하여라. 이 함수는 2개의 인수를 가진다. 첫번째 인수는 수들을 포함하는 배열에 대한 포인터이고, 두번째 인수는 배열의 크기를 나타내는 정수값이다. 이 함수를 사용하기 위한 프로그램도 작성하여라.

```
Project1 (전역 범위)
1  #include <iostream>
2  #include <Windows.h>
3  using namespace std;
4
5  double avg(double *arr, int size);
6
7  int main() {
8
9      double array[3] = { 1.234, 2.345, 3.456 };
10
11     for (int i = 0; i < 3; i++) {
12         cout << i + 1 << "번째 소수 : " << array[i] << endl;
13     }
14
15     cout << "소수 값들의 평균 : " << avg(array, 3) << endl;
16     // avg(array, 3)으로 인해 구해진 array에 들어있는 소수들의 평균값 출력
17
18     system("pause");
19 }
20
21 double avg(double *arr, int size) {
22
23     double avg = 0.0;
24
25     if (size <= 0) { // 전달된 것이 없거나 음수이면 0을 리턴
26         return avg;
27     }
28
29     for (int i = 0; i < size; i++) { // arr에 들어있는 모든 값을 더한다
30         avg += arr[i];
31     }
32
33     return avg / size; // 모두 더한값을 size로 나누어서 평균을 구하고 리턴
34 }
```

C:\Users\Owner\Desktop\단국대학교\객체지향프로그래밍\C++\SelfStudy\Project1\Debug\Project1.exe

```
1번째 소수 : 1.234
2번째 소수 : 2.345
3번째 소수 : 3.456
소수 값들의 평균 : 2.345
계속하려면 아무 키나 누르십시오 . . .
```

과제 2번

문자열을 뒤집는 `rev_str()` 함수를 작성하여라. `rev_str()` 함수가 한 개 혹은 두 개의 문자열을 처리할 수 있게 함수를 중복하여라. 한 개의 문자열로 호출되면, 문자열을 뒤집어서 그 문자열에 넣어서 반환하고, 두 개의 문자열로 호출되면 문자열들을 뒤집은 후 두번째 인수의 문자열에 넣어서 반환한다. 예를 들면 다음과 같다.

```
char s1[80], s2[80];
strcpy(s1, "hello");
rev_str(s1, s2); // 뒤집어진 문자열은 s2에 저장되어 변환되며,
                // s1은 변하지 않는다.
rev_str(s1); // 뒤집어진 문자열은 s1에 저장되어 반환된다.
```

```
1  #include <iostream>
2  #include <Windows.h>
3  using namespace std;
4
5  void rev_str(char *sen1);
6  void rev_str(char *sen1, char *sen2);
7
8  int main() {
9
10     char arr1[50]; // 값을 입력하고 저장할 2개의 배열 선언
11     char arr2[50];
12
13     cin >> arr1; // 문자열을 입력받는다
14     rev_str(arr1); // 입력된 문자열을 뒤집는다
15
16     cout << "rev_str(arr1) 실행" << endl;
17     cout << "arr1 : " << arr1 << endl;
18     // ex) programming을 입력하면 rev_str(arr1)에 의하여
19     // gnimmargorp가 arr1에 저장되어 출력
20
21     rev_str(arr1, arr2); // arr1에 저장된 값을 뒤집어서 arr2에 저장
22                         // arr1에 저장된 값은 변함이 없다
23                         // rev_str(arr1)에서 값이 뒤집어져 arr1에 저장되었기 때문에
24                         // arr2에는 다시 뒤집어져 원래의 문자열 출력
25
26     cout << "rev_str(arr1, arr2) 실행" << endl;
27     cout << "arr1 : " << arr1 << endl;
28     cout << "arr2 : " << arr2 << endl;
29     // ex) rev_str(arr1, arr2)에 의하여 arr1에 저장된 gnimmargorp가
30     // 뒤집혀 programming가 arr2에 저장되어 출력
31
32     system("pause");
33 }
34
35 void rev_str(char *sen1) {
36     char temp; // 값을 저장할 변수 선언
37     int len = strlen(sen1); // sen1에 저장된 문자열의 길이를 len에 저장
38
39     for (int i = 0; i < len / 2; i++) { // 앞뒤로 바꿔줄 것이므로 len / 2까지
40         temp = sen1[i]; // temp에 sen1[i] 저장
41         sen1[i] = sen1[len - 1 - i]; // 뒤에 것을 앞에 저장
42         sen1[len - 1 - i] = temp; // temp에 저장된 앞에 것을 뒤에 저장
43     }
44 }
45
46 void rev_str(char *sen1, char *sen2) {
47     int len = strlen(sen1); // sen1에 저장된 문자열의 길이를 len에 저장
48
49     for (int i = 0; i < len; i++) {
50         sen2[i] = sen1[len - 1 - i];
51         // sen2에 sen1의 뒤의 값부터 역순으로 저장하여 문자열 뒤집기
52     }
53
54     sen2[len] = '\0'; // 문자열의 다음 인덱스에 널값 저장
55 }
56
57 }
```

C:\Users\Owner\Desktop\단국대학교\객체지향프로그래밍\C++SelfStudy\Project1\Debug\Project1.exe

```
programming
rev_str(arr1) 실행
arr1 : gnimmargorp
rev_str(arr1, arr2) 실행
arr1 : gnimmargorp
arr2 : programming
계속하려면 아무 키나 누르십시오 . . .
```