

### reverseAr1D

Write a C function `reverseAr1D()` that takes in an array of integers `ar` and an integer `size` as parameters. The parameter `size` indicates the size of the array to be processed. The function converts the content in the array in reverse order and passes the array to the calling function via call by reference.

```
void reverseAr1D(int ar[ ], int size);
```

A sample program template is given below for testing the function:

```
#include <stdio.h>
void reverseAr1D(int ar[], int size);
int main()
{
    int ar[10];
    int size, i;

    printf("Enter array size: \n");
    scanf("%d", &size);
    printf("Enter %d data: \n", size);
    for (i=0; i <= size-1; i++)
        scanf("%d", &ar[i]);
    reverseAr1D(ar, size);
    printf("reverseAr1D(): ");
    if (size > 0) {
        for (i=0; i<size; i++)
            printf("%d ", ar[i]);
    }
    return 0;
}
void reverseAr1D(int ar[], int size)
{
    /* Write your code here */
}
```

Some sample input and output sessions are given below:

- (1) Test Case 1:  
Enter array size:  
5  
Enter 5 data:  
1 2 3 6 7  
reverseAr1D(): 7 6 3 2 1
- (2) Test Case 2:  
Enter array size:  
1  
Enter 1 data:  
5  
reverseAr1D(): 5
- (3) Test Case 3:  
Enter array size:

```
7
Enter 7 data:
1 2 3 4 5 6 7
reverseArlD(): 7 6 5 4 3 2 1
```

(4) Test Case 4:

```
Enter array size:
2
Enter 2 data:
2 4
reverseArlD(): 4 2
```