CHARACTER ARRAY

An array of characters is often referred to as string. A string is a sequence of characters treated as a single entity and is terminated by a null character ('\0'). The null character (‘\0’) is automatically added by the compiler at the end of the string. A string is actually a one-dimensional array of characters in C language.

Syntax for declaration:

char arrayName[arraySize];

For example,

```
char str1[SIZE];
```

Initializing an Array of Characters

There are three ways to initialize an array:

1. ***Initialize with the whole string***

For example,

```
char str1[SIZE] = "hello";
```

Individual characters of the string “hello” will be assigned to the memory locations. The compiler automatically adds the ‘\0’ at the end.

To illustrate:

0	1	2	3	4	5	6	7	8	9
h	e	l	L	o	\0				

2. ***Initialize with the individual characters***

When initializing the array with the individual characters, always add the null character (‘\0’) at the end.

For example,

```
char str1[SIZE] = {'h', 'e', 'l', 'l', 'o', '\0'};
```

3. ***Initialize using an input command***

An array of characters can be initialized using an input command, like gets().

For example,

```
char str1[SIZE];
gets(str1);
```

When using an input command, the string is input as a whole, thus ‘\0’ is automatically added by the compiler at the end.

Note that the null character is not included in counting the number of characters in a string. The null character terminates the string or just indicates that it is already the end of the string.

Output a Character Array

A character array can be output as a whole just like the example below:

```
printf("%s",str1);
```

Or as individual characters just like any other type of array. However, for a character array, the null character is used to terminate the loop instead of the size or count (i.e. the variable used to keep track the current number of elements.)

For example,

```
for(i=0;str1[i]!='\0';i++)
    printf("%c ",str1[i]);
```

Passing a Character Array to a Function

Since the null character is used to terminate an array of characters, the size parameter, usually used in passing arrays to functions becomes unnecessary.

Here is an example,

main.c | chararray.h | chararray.c

```
1  void display(char str1[]);
2  int countAs(char str1[]);
```

main.c | chararray.h | chararray.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void display(char str1[]){
5      int i;
6      for(i=0;str1[i]!='\0';i++)
7          printf("%c ",str1[i]);
8      printf("\n");
9  }
10
11 int countAs(char str1[]){
12     int i, count=0;
13     for(i=0;str1[i]!='\0';i++){
14         if(str1[i]=='a' || str1[i]=='A')
15             count++;
16     }
17     return count;
18 }
```

main.c | chararray.h | chararray.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "chararray.h"
#define SIZE 10

/* run this program using the console pauser o

int main(int argc, char *argv[]) {
    char str1[SIZE];
    gets(str1);
    display(str1);
    printf("Number of As: %d",countAs(str1));
    return 0;
}
```

```
America
A m e r i c a
Number of As: 2
```

Practice Exercise (Ungraded)

Define the function palindrome() that returns 1 if the parameter is a palindrome, and 0 if otherwise. A palindrome is a word, phrase, or sequence that reads the same backward as forward.

```
int palindrome(char str1[]);
```