strlen() implementation

```
/* strlen : return length of string s */
int strlen(char *s)
{
    int n;

    for (n = 0 ; s[n] != '\0' ; n++) ;
    return n;
}
```

```
/* strlen : return length of string s */
int strlen(char *s)
{
    int n;

    for (n = 0 ; *s != '\0' ; s++) n++;
    return n;
}
```

strcpy() implementation

```
/* strcpy : copy t to s */
void strcpy(char *s, char *t)
{
    int i=0;
    while ((s[i] = t[i]) != ‘\0’) i++;
}
```

```
/* strcpy : copy t to s */
void strcpy(char *s, char *t)
{
    while ((*s = *t) != ‘\0’) {
        s++;
        t++;
    }
}
```

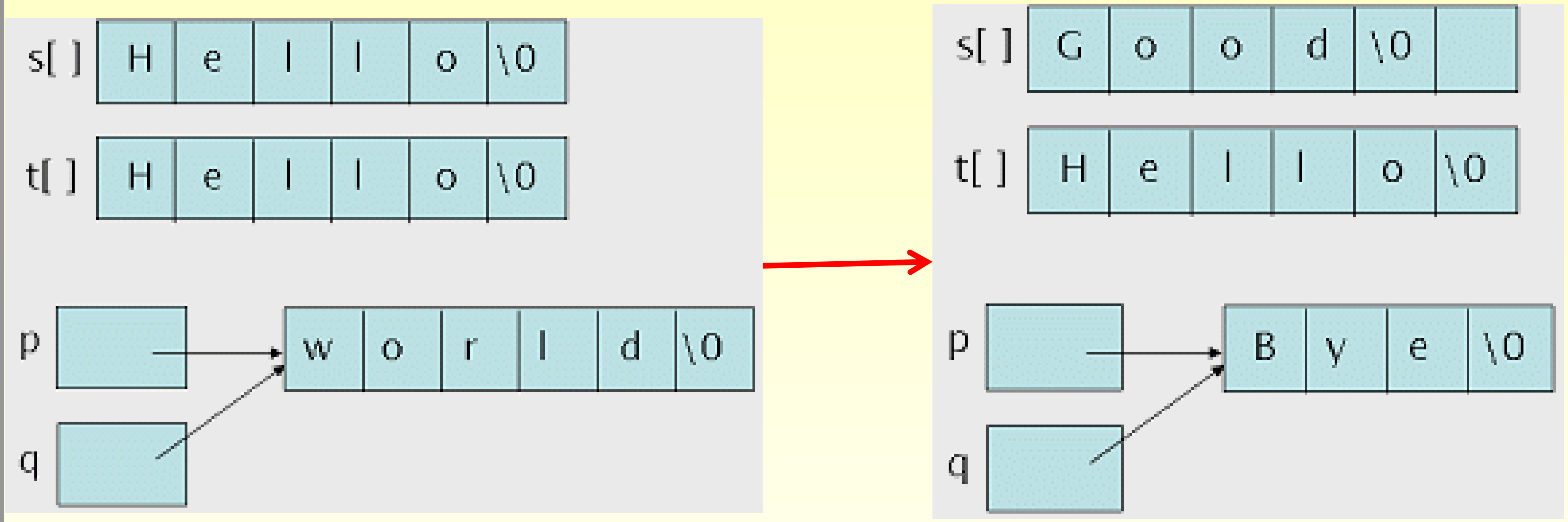
```
/* strcpy : copy t to s */
void strcpy(char *s, char *t)
{
    while (*s++ = *t++) ;
}
```

strcmp() implementation

```
/* strcmp : return <0 if s<t, 0 if s==t, >0 if s>t
int strcmp(char *s, char *t)
{
    int i;

    for (i = 0; s[i] == t[i]; i++)
        if (s[i] == '\0') return 0;
    return s[i] - t[i];
}
```

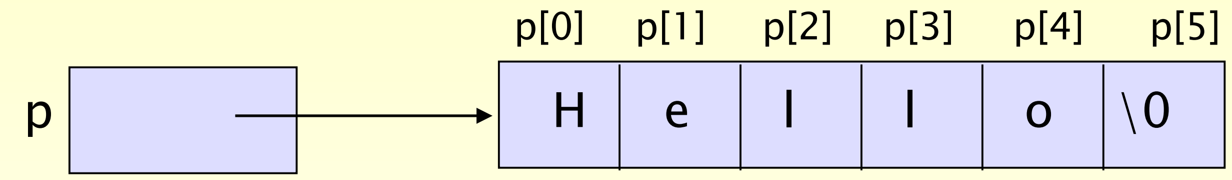
```
int strcmp(char *s, char *t)
{
    for ( ; *s == *t; s++, t++)
        if (*s == '\0') return 0;
    return *s - *t;
}
```

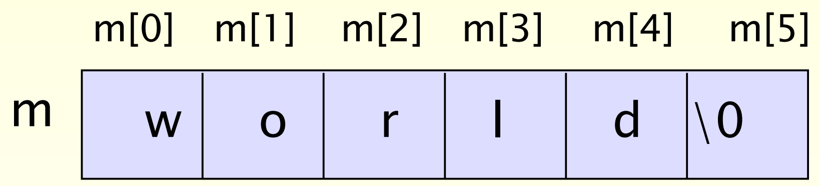
```
#include <stdio.h>
#include <string.h>
#define MAX_LINE 81

int main()
{
        char s[] = "Hello", t[6];
        char *p = "world", *q;

        printf(“string s = %s\n", s);
        strcpy(t,s);
        printf(“string t = %s\n", t);

        printf(“string p = %s\n", p);
        q = p;
        printf(“string q = %s\n", q);

        strcpy(s, "Good");
        printf(“string s = %s\n", s);
        printf(“string t = %s\n", t);

        strcpy(p, "Bye");
        printf(“string p = %s\n", p);
        printf(“string q = %s\n", q);

        return 0;
}
```


s[ ]
H e l l o \0
t[ ]
H e l l o \0
p
q
w o r l d \0
s[ ]
G o o d \0
t[ ]
H e l l o \0
p
q
B y e \0

String

```
char *p ="Hello";
char m[]="world";
```





String input

- Example1)

```
char *name;
scanf("%s", name);
```

- Example2)

```
char name[81];
scanf("%s", name);
```

- Example3)

```
char *name;
name=(char*)malloc(sizeof(char)*81);
scanf("%s", name);
…

free(name); // deallocate when name is no longer useful
```

Multiple String

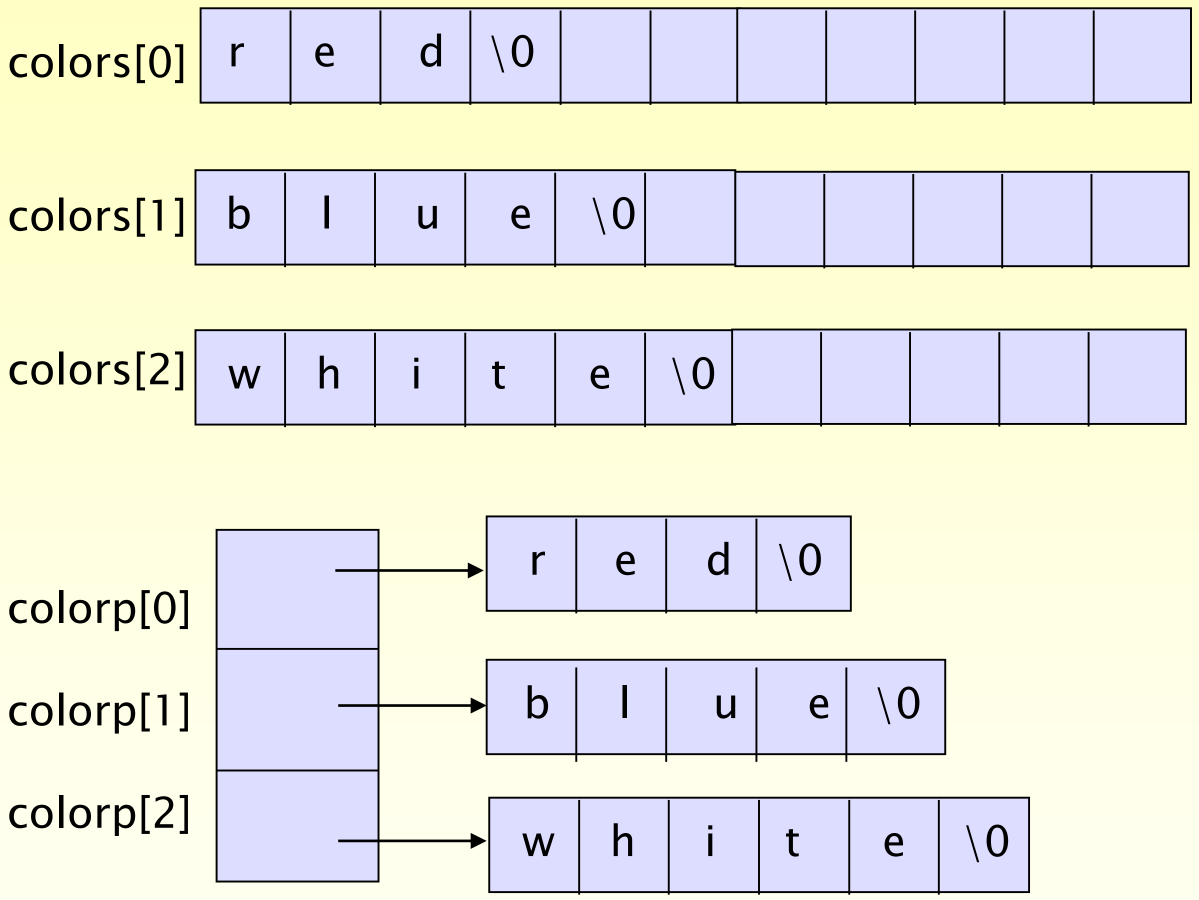
- Using 2 dimensional array

```
char colors[3][10]= {"red", "blue", "white"};
```

or

```
char *colorp[3] = {"red", "blue", "white"};
```

Multiple String



String Input/Output

- **char *gets(char *str);**
  - Read one line string from keyboard
  - Put the input string into str

- **int puts(char *str);**
  - Print string str into standard output

String Input/Output

- int sprintf(char *str, char *format, ...);
  - Put the output into str instead of standard output

- int sscanf(char *str, char *format, ...);
  - Get the input from str instead of standard input

```c
#include<stdio.h>
#define MAX_LINE 81
#define MAX_WORD 21

int main()
{
    char str1[MAX_LINE]="C programming", str2[MAX_LINE]="language.";
    char temp[MAX_LINE];

    puts(str1);
    puts(str2);
    printf("%s", str1);
    printf("%s\n", str2);

    sprintf(temp, "%s %s is beautiful\n", str1, str2);
    printf("%s", temp);

    return 0;
}
```

```
Output :
C programming
language
C programming language
C programming language is beautiful
```

Other String functions

- strcpy, strcat, strcmp, strlen

- int atoi(char *str);   // ascii to integer
- double atof(char *str); // ascii to double

- char *strstr(char *str1, char *str2);

  // search for str2 in str1

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_LINE 81

int main()
{
         float sum = 0;
         int count = 0;
         char num[MAX_LINE];

         printf("get price : \n");
         while (gets(num) != NULL) {
                  count++;
                  sum = sum + atof(num);
         }
         printf(" %d items , Sum : %6.2f \n", count, sum);

         return 0;
}
```

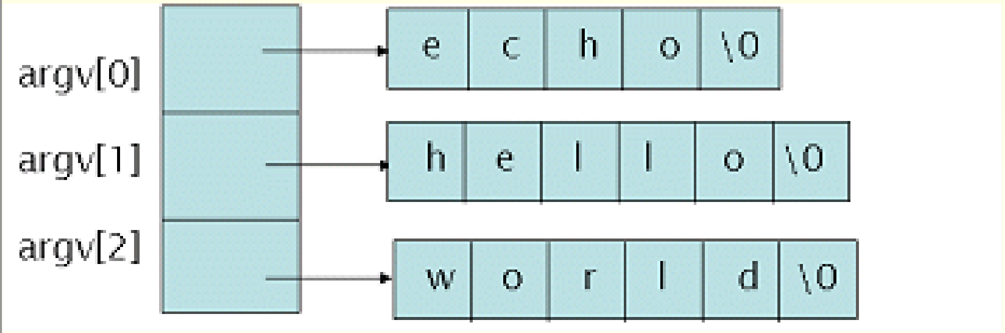
```
Output :

get price :
15.5
31.40
180.05
29.99
^Z
4 items , Sum : 256.99
```

Main arguments

- int main(int argc, char *argv[])
- argc : number of arguments
- argv
  - argv[0] : execution file name
  - argv[1] : first argument string
  - argv[2] : second argument string
  - …

```
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[])
{
    int i;

    for(i = 1;  i< argc;  i++)
        printf("%s%s", argv[i], (i < argc - 1) ? " " : " \n");

    return 0;
}
```

C\:> echo hello world



실행결과:

```
C:> echo hello world
hello world
```