

برنامه‌سازی کامپیووتر

جلسه نهم

رشته‌ها در زبان C

طرح کلی

- آرایه‌های کاراکتری (رشته‌ها)
 - تعریف رشته‌ها
 - مقداردهی اولیه در رشته‌ها
 - ورودی و خروجی رشته‌ها
 - رشته‌ها به عنوان پارامتر توابع
 - سرفایل string.h
 - انتساب رشته‌ها
 - مقایسه دو رشته
 - الحق دو رشته
 - آرایه‌ای از رشته‌ها

مقدمه

- در زبان C رشته بصورت آرایه‌ای از کاراکترها تعریف می‌شود.
- رشته‌ها برای ذخیره، بازیابی و دستکاری متن‌ها مورد استفاده قرار می‌گیرند.
- برای تعیین انتهای رشته از کاراکتر خاصی به نام (تهی=NULL) استفاده می‌شود که با '\0' مشخص می‌گردد. بنابراین، آخرین کاراکتر در هر رشته برابر با '\0' می‌باشد.
- لذا اگر رشته‌ای به طول n تعریف کنیم، فقط از n-1 کاراکتر آن می‌توانیم استفاده کنیم، زیرا کاراکتر آخر '\0' است.

مثال:

```
char s[10];
```

اگر محتویات این رشته برابر با "sahand" قرار گیرد، این رشته به صورت زیر نمایش داده می‌شود:

s	s	a	h	a	n	d	\0	?	?	?
	s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]

پس همواره طول رشته را یکی بیشتر از طول مورد نیاز در نظر بگیرید.

مقداردهی اولیه به رشته‌ها(۱)

- هنگام **تعريف** رشته‌ها می‌توان به آن‌ها مقدار اولیه داد.
 - هنگام مقداردهی اولیه می‌توان طول رشته را مشخص نکرد. در این صورت، اندازه آرایه یک واحد بیشتر از تعداد کاراکترهایی است که به آن نسبت داده می‌شود.
 - دو روش برای مقداردهی اولیه به رشته‌ها وجود دارد:
روش اول:
- رشته در داخل کوتشن قرار گیرد و به متغیر رشته‌ای (آرایه رشته‌ای) نسبت داده شود:

char s[] = “sahand”;

s	s	a	h	a	n	d	\0
	s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]

char s[12] = “Tabriz”;

s	T	a	b	r	i	z	\0	?	?	?	?
	s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]	s[10]

در این روش کاراکتر ‘\0’ به‌طور خودکار به انتهای رشته اضافه می‌شود.

مقداردهی اولیه به رشته‌ها(۲)

روش دوم:

هر یک از کاراکترهای رشته‌ای به عنوان یک عنصر رشته به آرایه نسبت داده شوند:

char s [] = {'A', 'l', 'i', '\0'};

s	A	l	i	\0
	s[0]	s[1]	s[2]	s[3]

در این روش '\0' باید توسط برنامه‌نویس در انتهای رشته قرار داده شود.

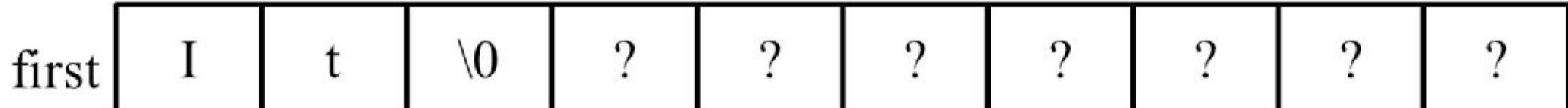
ورودی و خروجی رشته‌ها

- برای ورودی و خروجی رشته‌ها می‌توان از توابع `printf()` و `scanf()` استفاده کرد.
- نباید از عملگر آدرس & قبل از نام آرایه استفاده کرد. زیرا نام آرایه خود نشان‌دهنده‌ی آدرس است.

```
char first[10];
scanf("%s", first);
printf("\nFirst string is : %s", first);
```

It is . . .

First string is : It



نکته: خواندن با `(scanf()` تا رسیدن به اولین فضای خالی، `ENTER` یا `Tab` پیدا می‌کند.

- علاوه بر این دو تابع، توابع زیر نیز از سرفایل `stdio.h` مورد استفاده قرار می‌گیرند:
 - تابع ورودی `(gets()`
 - تابع خروجی `(puts()`

تابع gets()

- خواندن رشته با تابع gets() انجام می‌گیرد.
- این تابع در سرفایل stdio.h قرار دارد.
- این تابع بعد از خواندن رشته، سطر جاری را رد می‌کند.
- نحوه کاربرد آن بصورت زیر است:

gets(متغیر رشته‌ای);

مثال:

```
char str[21];
gets(str);
```

- پس از وارد کردن رشته باید کلید ENTER را نیز فشار داد.

تفاوت `scanf()` و `gets()`

- تفاوت `scanf()` و `gets()` در این است که در تابع `scanf()` فقط کلید ENTER انتهای رشته را مشخص می‌کند. لذا رشته می‌تواند حاوی فاصله (space) و یا Tab باشد. در حالی که در تابع `gets()` فاصله و Tab نیز به عنوان جداگانه تلقی شده، انتهای رشته را مشخص می‌کند.

```
char s[20];
scanf("%s", s);
```

Sahand University!

s	S	a	h	a	n	d	\0	?	?	?	?	?	?	?	?	?	?	?	?	?
	s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]	s[10]	s[11]	s[12]	s[13]	s[14]	s[15]	s[16]	s[17]	s[18]	s[19]

```
gets(s);
```

s	S	a	h	a	n	d		U	n	i	v	e	r	s	i	t	y	!	\0	?
	s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]	s[10]	s[11]	s[12]	s[13]	s[14]	s[15]	s[16]	s[17]	s[18]	s[19]

تابع puts()

- چاپ رشته با تابع puts() انجام می‌گیرد.
- این تابع در سرفایل stdio.h قرار دارد.
- این تابع پس از نوشتن رشته، سطر جاری را رد می‌کند.

کاربرد آن به صورت زیر است:

```
puts("رشته");  
puts(متغیر رشته‌ای);
```

مثال:

```
char name[] = "ali";  
puts("Your name is: ");  
puts(name);
```

```
Your name is:  
ali
```

مثال

برنامه‌ای بنویسید که رشته‌ای را دریافت کرده و در آرایه‌ای کاراکتری ذخیره کند. سپس طول رشته را محاسبه کرده و چاپ کند.

```
#include <stdio.h>
int main(){
    int i=0;
    char str[50];

    gets(str);

    while(str[i]!='\0')
        i++;

    printf("Length of your string is : %d", i);
    getch();
    return 0;
}
```

It is a statement.
Length of your string is : 18

رشته‌ها به عنوان آرگومان توابع

```
void func1 (char x[]);
void func2 (char x[], int len);
int main()
{
    char x[10];
    ...
    func1(x);
    ...
    func2(x, 10);
    return 0;
}
```

```
void func1 (char x[10])
```

```
{
```

```
    ...
```

```
}
```

```
void func2 (char x[], int len)
```

```
{
```

```
    ...
```

```
}
```

- چون رشته‌ها به صورت آرایه‌ای از کاراکترها تعریف می‌شوند، شیوه ارسال رشته‌ها به توابع، همانند آرایه است.

- در آرگومان تابع، نام رشته ذکر می‌شود.

- پارامتر معادل با رشته می‌تواند یکی از سه مورد زیر باشد:

- آرایه‌ای با طول معین

- آرایه‌ای با طول نامعین

- اشاره‌گر

مثال

برنامه‌ای بنویسید که رشته‌ای را خوانده، تمام حروف کوچک آن رشته را به حروف بزرگ تبدیل کرده و چاپ می‌کند.

```
#include <stdio.h>
void upper(char []);
int main()
{
    char s[21];
    printf("Enter a string: ");
    gets(s);
    upper(s);
    puts("Result is:");
    puts(s);
    getch();
    return 0;
}
```

Enter a string: **It is a string.**
Result is:
IT IS A STRING

```
void upper(char s[21])
{
    int i;
    for (i=0;i<21;i++)
        if (s[i]>='a' && s[i]<='z')
            s[i]-=32;
}
```

مثال

برنامه‌ای بنویسید که یک رشته و دو کاراکترهای ch1 و ch2 را خوانده، تمام کاراکترهای ch1 را در رشته به ch2 تبدیل می‌کند.

```
#include <stdio.h>
void replace(char [], char, char);
int main()
{
    char string[50];
    char source_letter,target_letter;
    printf("Enter a string: ");
    gets(string);
    printf("\nEnter source character: ");
    source_letter=getche();
    printf("\nEnter target character:");
    target_letter=getche();

    replace(string, source_letter,target_letter);

    puts("\nResult is:");
    puts(string);

    getch();
    return 0;
}
```

Enter a string: Ali is a manager.

Enter source character: a

Enter target character: t

Result is:

Ali is t mtntger.

```
void replace(char s[], char ch1, char ch2)
{
    int i;
    for (i=0;i<50;i++)
        if (s[i]==ch1)
            s[i]=ch2;
}
```

string.h سرفایل

- این سرفایل شامل بسیاری از توابع پرکاربرد برای رشته‌ها می‌باشد.
 - انتساب رشته‌ها `strcpy(s1,s2)`
 - مقایسه دو رشته `strcmp(s1,s2)`
 - الحاقد دو رشته `strcat (s1, s2)`

انتساب رشته‌ها (۱)

- انتساب رشته‌ها به معنی کپی کردن یک رشته در رشته دیگر است.
- با استفاده از عملگر = نمی‌توان رشته‌ها را انتساب کرد. یعنی اگر s1 و s2 آرایه‌هایی کاراکتری (رشته) باشند، استفاده از دستور زیر نادرست است:

```
s1 = "Sahand";  
s2 = s1;
```

- برای کپی کردن رشته‌ای در رشته دیگر و یا انتساب رشته‌ای به رشته دیگر، از تابع strcpy() استفاده می‌شود.
- این تابع در سرفایل string.h قرار دارد و بصورت زیر بکار می‌رود:

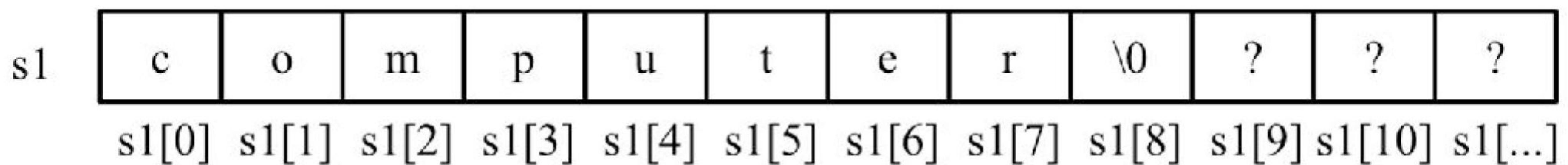
```
strcpy (str1 , str2);
```

- str2 می‌تواند رشته‌ای از کارکترها (مانند "Sahand University") و یا یک متغیر رشته‌ای باشد.
- با اجرای این دستور، مقادیری که در str2 قرار دارند به str1 کپی می‌شوند.

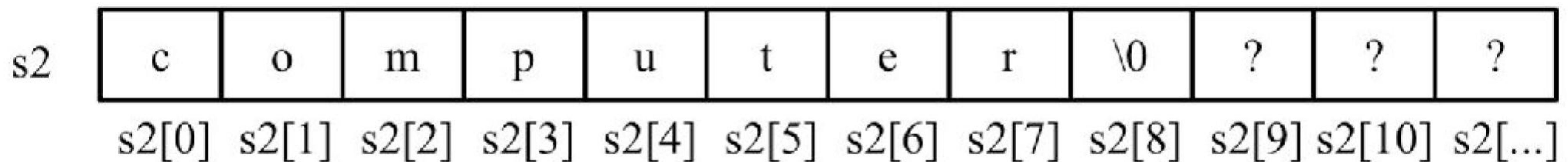
انتساب رشته‌ها (۲)

: مثال

```
strcpy(s1,"computer");
```



```
strcpy(s2,s1);
```



- اگر طول str2 بیش از طول str1 باشد، تا جایی که عمل کپی کردن امکان‌پذیر باشد، عمل کپی در آن رشته صورت می‌گیرد و کاراکترهای اضافی بلافاصله پس از str1 قرار می‌گیرند. در نتیجه، اگر متغیرهایی بعد از str1 قرار داشته باشند، محتویات آنها از بین خواهد رفت.

```
char s[3];
```



```
strcpy(s,"This is a statement.");
```

مقایسه رشته‌ها (۱)

- اگر بخواهیم دو متغیر صحیح X و y را با هم مقایسه کنید، از دستور $if(x==y)$ استفاده می‌کنیم. اما برای مقایسه دو رشته $s1$ و $s2$ نمی‌توان از $if(s1==s2)$ استفاده کرد.
- برای مقایسه رشته‌ها از تابع `strcmp()` استفاده می‌شود.
- این تابع در سرفایل `string.h` قرار دارد. بصورت زیر بکار می‌رود:
`strcmp(s1,s2)`
- حاصل کار این تابع یک عدد است که مقدار آن بیانگر وضعیت دو رشته نسبت به هم است. اگر عدد برگردانده شده توسط تابع:
 - برابر با صفر باشد، $s1=s2$ می‌باشد.
 - منفی باشد، $s1 < s2$ می‌باشد.
 - مثبت باشد، $s1 > s2$ می‌باشد.

مقایسه رشته‌ها (۲)

- منظور از مقایسه رشته‌ها، مقایسه کاراکتری آن‌ها می‌باشد:

اولین کاراکتر از رشته اول با اولین کاراکتر از رشته دوم مقایسه می‌شود. اگر مساوی باشند، کاراکترهای بعدی با هم مقایسه می‌شوند.

اگر تمام کاراکترهای دو رشته با هم مساوی باشند، آن دو رشته با هم مساویند. با رسیدن به اولین مورد اختلاف، کاراکتری که بزرگتر است (یعنی کد اسکی بزرگتری دارد)، رشته‌ی حاوی آن کاراکتر، رشته بزرگتر خواهد بود.

s1	s2	نتیجه مقایسه
“Sahand”	“Sahand”	$s1=s2$
“SahandUni”	“SahandCity”	$s1>s2$
“abcd...”	“abce...”	$s1<s2$

الحق دو رشته (۱)

- الحق کردن دو رشته به معنی قرار دادن یک رشته در انتهای رشته دیگر است.

مثال : اگر $s1$ برابر با "University" و $s2$ برابر با "Sahand" باشد، الحق $s2$ به انتهای $s1$ موجب می‌شود تا $s1$ حاوی رشته "Sahand University" شود.

- تابع ([strcat\(\)](#)) که در سرفایل `string.h` قرار دارد برای الحق دو رشته بکار می‌رود.

```
strcat (s1, s2);
```

- با این دستور، $s2$ در انتهای رشته‌ی $s1$ قرار می‌گیرد.
- $s2$ می‌تواند متغیر رشته‌ای و یا رشته‌ای دلخواه (مانند "Sahand") باشد.
- $s1$ همواره متغیری رشته‌ای (آرایه‌ای کاراکتری) است.

الحاق دو رشته (۲)

```
char s1[21] = "Sahand";
char s2[21] = " University";
strcat(s1,s2);
```

مثال:

s1	S	a	h	a	n	d	\0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	s1[0]	s1[1]	s1[2]	s1[3]	s1[4]	s1[5]	s1[6]	s1[7]	s1[8]	s1[9]	s1[10]	s1[11]	s1[12]	s1[13]	s1[14]	s1[15]	s1[16]	s1[17]	s1[18]	s1[19]	s1[20]	
s2		U	n	i	v	e	r	s	i	t	y	\0	?	?	?	?	?	?	?	?	?	?
	s2[0]	s2[1]	s2[2]	s2[3]	s2[4]	s2[5]	s2[6]	s2[7]	s2[8]	s2[9]	s2[10]	s2[11]	s2[12]	s2[13]	s2[14]	s2[15]	s2[16]	s2[17]	s2[18]	s2[19]	s2[20]	
s1	S	a	h	a	n	d		U	n	i	v	e	r	s	i	t	y	\0	?	?	?	?
	s1[0]	s1[1]	s1[2]	s1[3]	s1[4]	s1[5]	s1[6]	s1[7]	s1[8]	s1[9]	s1[10]	s1[11]	s1[12]	s1[13]	s1[14]	s1[15]	s1[16]	s1[17]	s1[18]	s1[19]	s1[20]	

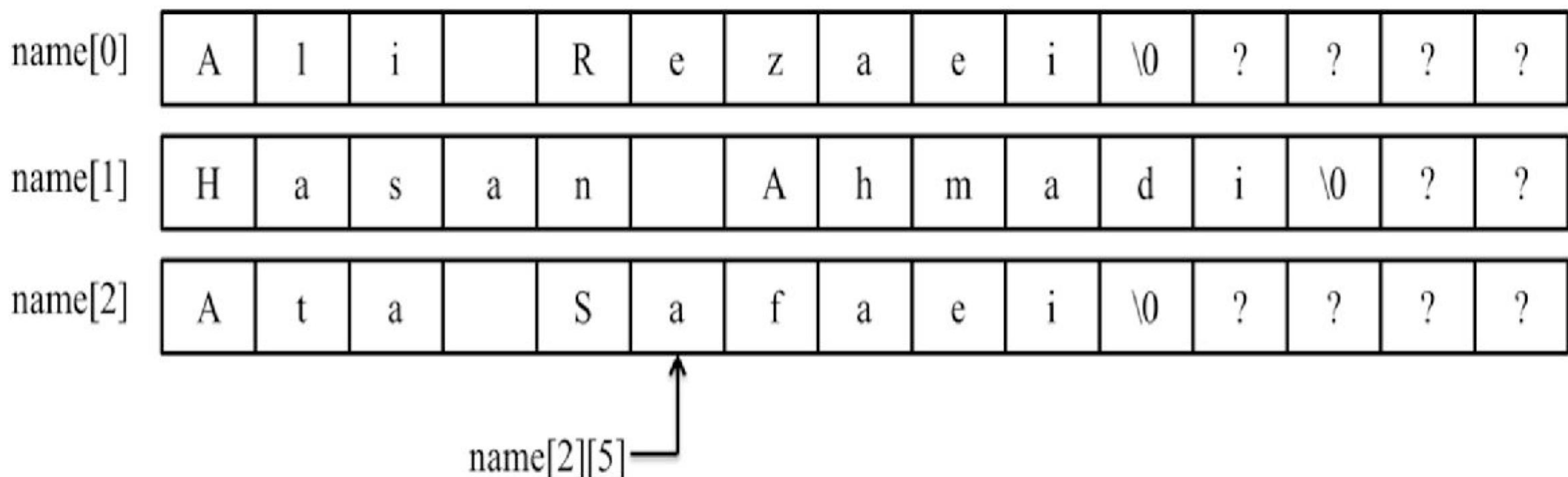
- چنانچه طول رشته s1 طوری باشد که گنجایش s2 را نداشته باشد، بقیه رشته s2 در ادامه رشته s1 قرار می‌گیرد و در نتیجه چنانچه متغیرهایی بعد از s1 وجود داشته باشند، محتویات آنها از بین می‌رود.

s1	S	a	h	a	n	d		U	n	i	v	e	r	s	i	t	y	\0					
	s1[0]	s1[1]	s1[2]	s1[3]	s1[4]	s1[5]	s1[6]	s1[7]	s1[8]	s1[9]	s1[10]	s1[11]	s1[12]	s1[13]	s1[14]	s1[15]	s1[16]	s1[17]					

آرایه‌ای از رشته‌ها

- در مواردی که می‌خواهیم تعداد زیادی رشته را تعریف کرده و مورد استفاده قرار دهیم، از آرایه‌ای از رشته‌ها استفاده می‌کنیم. برای مثال اگر بخواهیم اسمی دانشجویان یک کلاس را ذخیره کنیم.
 - آرایه‌ای از رشته‌ها یعنی همان آرایه‌ای از آرایه‌ها که همان آرایه دو بعدی است.
 - پس می‌توان آرایه‌ای دو بعدی تعریف کرد که بعد اول تعداد رشته‌ها، و بعد دوم طول هر رشته را نشان می‌دهد.
- مثال: اگر بخواهیم نام ۳ دانشجو را که طول هر نام بیش از ۱۴ کاراکتر نیست، ذخیره کنیم، از آرایه name بصورت زیر استفاده می‌کنیم.

```
char name[3][15];
```



مثال

برنامه‌ای بنویسید که سه نام را به ترتیب دریافت کرده و اسامی را از آخر به اول چاپ کند.

```
#include <stdio.h>
int main(){
    int i;
    char name[3][15];
    for (i=0;i<3;i++)
        gets(name[i]);
    puts("***** Inverse of names: *****");
    for (i=2;i>=0;i--)
        puts(name[i]);
    getch();
    return 0;
}
```

Ali
Mohammad
Amin
***** Inverse of names: *****
Amin
Mohammad
Ali

تمرینات تکمیلی

برنامه‌های زیر را به ترتیب با استفاده از یکی از کامپایلرهای زبان C طراحی، پیاده‌سازی و اجرا کنید. (در تمامی سوالات فرض کنید طول رشته‌ها بیش از ۵۰ کاراکتر نمی‌شود.)

1. برنامه‌ای بنویسید که جمله‌ای را دریافت کرده و در یک رشته ذخیره کند. سپس کلمات رشته را جدا کرده و به ترتیب چاپ کند.
2. برنامه‌ای بنویسید که دو رشته را دریافت کرده ونتیجه مقایسه این دو رشته را با پیغامی مناسب نمایش دهد.
3. تابعی بنویسید که آرایه‌ای کاراکتری (رشته) را گرفته و طول واقعی آن را برگرداند. (آخرین عنصر هر رشته برابر با تهی '0' است).
4. برنامه‌ای بنویسید که رشته‌ای را از ورودی گرفته در آرایه‌ای ذخیره کند و آن را بصورت معکوس نمایش دهد.
5. تابعی بنویسید که یک رشته و یک کاراکتر ch1 را گرفته، و تعداد کاراکترهای ch1 موجود در رشته را شمرده و تعداد آن‌ها را بازگرداند.
6. برنامه‌ای بنویسید که عدد ۲۰ رقمی را بصورت رشته‌ای از ارقام گرفته و در آرایه‌ای کاراکتری ذخیره کند.
7. دو عدد ۲۰ رقمی را از ورودی دریافت کرده و به ترتیب در دو رشته قرار دهید. تابعی بنویسید که این دو رشته را دریافت کرده و مجموع آن‌ها را در رشته‌ای دیگر ذخیره کرده و سپس نمایش دهد.
8. برنامه‌ای بنویسید که ۵ رشته را به ترتیب از ورودی گرفته و در آرایه‌ای از رشته‌ها ذخیره کند. تابعی بنویسید که این آرایه را گرفته و رشته‌ها را به ترتیب چاپ کند.

جمع‌بندی

- رشته‌ها
 - تعریف رشته‌ها
 - مقداردهی اولیه در رشته‌ها
 - ورودی و خروجی رشته‌ها
 - رشته‌ها به عنوان پارامتر توابع
 - سرفایل string.h
 - انتساب رشته‌ها
 - مقایسه دو رشته
 - الحاق دو رشته
 - آرایه‌ای از رشته‌ها