



اصول برنامه نویسی کامپیوتر

رشته ها:

محمد فرشی

آزمایشگاه الگوریتم های ترکیبیاتی و هندسی - دانشکده علوم ریاضی - دانشگاه یزد

۱۴۰۳-۱

رشته‌ها

بسیاری از داده‌ها نظیر نام و نام خانوادگی، عنوان درس و بسیاری موارد مشابه به صورت رشته، یعنی دنباله‌ای از کاراکترها هستند.

لذا لازم است امکان ذخیره‌سازی و پردازش رشته‌ها نیز در زبان برنامه‌نویسی موجود باشد.

در زبان C++ نوع داده‌ای رشته‌ای وجود ندارد و برای نگهداری رشته‌ها دو روش استفاده می‌شود.

یکی این که رشته‌ها را به صورت آرایه‌ای از کاراکترها در نظر می‌گیرد و دیگری استفاده از کتابخانه string است.

در این فصل به قسمت اول می‌پردازیم. برای اطلاع از روش دوم، خوانندگان را به بررسی کتابخانه مربوطه ارجاع می‌دهیم.



رشته‌ها

بسیاری از داده‌ها نظیر نام و نام خانوادگی، عنوان درس و بسیاری موارد مشابه به صورت رشته، یعنی دنباله‌ای از کاراکترها هستند.

لذا لازم است امکان ذخیره‌سازی و پردازش رشته‌ها نیز در زبان برنامه‌نویسی موجود باشد.

در زبان C++ نوع داده‌ای رشته‌ای وجود ندارد و برای نگهداری رشته‌ها دو روش استفاده می‌شود.

یکی این که رشته‌ها را به صورت آرایه‌ای از کاراکترها در نظر می‌گیرد و دیگری استفاده از کتابخانه string است.

در این فصل به قسمت اول می‌پردازیم. برای اطلاع از روش دوم، خوانندگان را به بررسی کتابخانه مربوطه ارجاع می‌دهیم.



رشته‌ها

بسیاری از داده‌ها نظیر نام و نام خانوادگی، عنوان درس و بسیاری موارد مشابه به صورت رشته، یعنی دنباله‌ای از کاراکترها هستند.

لذا لازم است امکان ذخیره‌سازی و پردازش رشته‌ها نیز در زبان برنامه‌نویسی موجود باشد.

در زبان C++ نوع داده‌ای رشته‌ای وجود ندارد و برای نگهداری رشته‌ها دو روش استفاده می‌شود.

یکی این که رشته‌ها را به صورت آرایه‌ای از کاراکترها در نظر می‌گیرد و دیگری استفاده از کتابخانه string است.

در این فصل به قسمت اول می‌پردازیم. برای اطلاع از روش دوم، خوانندگان را به بررسی کتابخانه مربوطه ارجاع می‌دهیم.



رشته‌ها

بسیاری از داده‌ها نظیر نام و نام خانوادگی، عنوان درس و بسیاری موارد مشابه به صورت رشته، یعنی دنباله‌ای از کاراکترها هستند.

لذا لازم است امکان ذخیره‌سازی و پردازش رشته‌ها نیز در زبان برنامه‌نویسی موجود باشد.

در زبان C++ نوع داده‌ای رشته‌ای وجود ندارد و برای نگهداری رشته‌ها دو روش استفاده می‌شود.

یکی این که رشته‌ها را به صورت آرایه‌ای از کاراکترها در نظر می‌گیرد و دیگری استفاده از کتابخانه string است.

در این فصل به قسمت اول می‌پردازیم. برای اطلاع از روش دوم، خوانندگان را به بررسی کتابخانه مربوطه ارجاع می‌دهیم.



رشته‌ها

بسیاری از داده‌ها نظیر نام و نام خانوادگی، عنوان درس و بسیاری موارد مشابه به صورت رشته، یعنی دنباله‌ای از کاراکترها هستند.

لذا لازم است امکان ذخیره‌سازی و پردازش رشته‌ها نیز در زبان برنامه‌نویسی موجود باشد.

در زبان C++ نوع داده‌ای رشته‌ای وجود ندارد و برای نگهداری رشته‌ها دو روش استفاده می‌شود.

یکی این که رشته‌ها را به صورت آرایه‌ای از کاراکترها در نظر می‌گیرد و دیگری استفاده از کتابخانه string است.

در این فصل به قسمت اول می‌پردازیم. برای اطلاع از روش دوم، خوانندگان را به بررسی کتابخانه مربوطه ارجاع می‌دهیم.



رشته‌ها

بسیاری از داده‌ها نظیر نام و نام خانوادگی، عنوان درس و بسیاری موارد مشابه به صورت رشته، یعنی دنباله‌ای از کاراکترها هستند.

لذا لازم است امکان ذخیره‌سازی و پردازش رشته‌ها نیز در زبان برنامه‌نویسی موجود باشد.

در زبان C++ نوع داده‌ای رشته‌ای وجود ندارد و برای نگهداری رشته‌ها دو روش استفاده می‌شود.

یکی این که رشته‌ها را به صورت آرایه‌ای از کاراکترها در نظر می‌گیرد و دیگری استفاده از کتابخانه string است.

در این فصل به قسمت اول می‌پردازیم. برای اطلاع از روش دوم، خوانندگان را به بررسی کتابخانه مربوطه ارجاع می‌دهیم.



رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

در این روش، برای نگهداری یک رشته، از یک آرایه از نوع `char` استفاده می‌کنیم (هر کاراکتر رشته در یک خانه از آرایه ذخیره می‌شود).

البته در حقیقت، به جای هر کاراکتر، کد اسکی آن نگهداری می‌شود.

ممکن است فضای آرایه، بیشتر از طول رشته باشد، لذا طبق قرارداد، در انتهای رشته، کاراکتر `'\0'` را قرار می‌دهیم.

دقت کنید که ثابت‌های رشته‌ای با " محدود می‌شدند ولی برای کاراکترها، هر کاراکتر بین دو ' قرار می‌گیرد، مثلاً کاراکتر `a` به صورت `'a'` بیان می‌شود.

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام `str` به صورت زیر عمل می‌کنیم:

```
char str[10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

str	'H'	'e'	'l'	'l'	'o'	'\0'					
-----	-----	-----	-----	-----	-----	------	--	--	--	--	--



رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

در این روش، برای نگهداری یک رشته، از یک آرایه از نوع char استفاده می‌کنیم (هر کاراکتر رشته در یک خانه از آرایه ذخیره می‌شود).

البته در حقیقت، به جای هر کاراکتر، کد اسکی آن نگهداری می‌شود.

ممکن است فضای آرایه، بیشتر از طول رشته باشد، لذا طبق قرارداد، در انتهای رشته، کاراکتر '\0' را قرار می‌دهیم.

دقت کنید که ثابت‌های رشته‌ای با " محدود می‌شدند ولی برای کاراکترها، هر کاراکتر بین دو ' قرار می‌گیرد، مثلاً کاراکتر a به صورت 'a' بیان می‌شود.

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام str به صورت زیر عمل می‌کنیم:

```
char str[10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

str	'H'	'e'	'l'	'l'	'o'	'\0'					
-----	-----	-----	-----	-----	-----	------	--	--	--	--	--



رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

- در این روش، برای نگهداری یک رشته، از یک آرایه از نوع `char` استفاده می‌کنیم (هر کاراکتر رشته در یک خانه از آرایه ذخیره می‌شود).
- البته در حقیقت، به جای هر کاراکتر، کد اسکی آن نگهداری می‌شود.
- ممکن است فضای آرایه، بیشتر از طول رشته باشد، لذا طبق قرارداد، در انتهای رشته، کاراکتر `'\0'` را قرار می‌دهیم.
- دقت کنید که ثابت‌های رشته‌ای با " محدود می‌شدند ولی برای کاراکترها، هر کاراکتر بین دو ' قرار می‌گیرد، مثلاً کاراکتر `a` به صورت `'a'` بیان می‌شود.
- مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام `str` به صورت زیر عمل می‌کنیم:

```
char str[10]={'H','e','l','l','o','\0'};
```

- به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

str	'H'	'e'	'l'	'l'	'o'	'\0'					
-----	-----	-----	-----	-----	-----	------	--	--	--	--	--



رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

در این روش، برای نگهداری یک رشته، از یک آرایه از نوع char استفاده می‌کنیم (هر کاراکتر رشته در یک خانه از آرایه ذخیره می‌شود).

البته در حقیقت، به جای هر کاراکتر، کد اسکی آن نگهداری می‌شود.

ممکن است فضای آرایه، بیشتر از طول رشته باشد، لذا طبق قرارداد، در انتهای رشته، کاراکتر '\0' را قرار می‌دهیم.

دقت کنید که ثابت‌های رشته‌ای با " محدود می‌شدند ولی برای کاراکترها، هر کاراکتر بین دو ' قرار می‌گیرد، مثلاً کاراکتر a به صورت 'a' بیان می‌شود.

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام str به صورت زیر عمل می‌کنیم:

```
char str[10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

str	'H'	'e'	'l'	'l'	'o'	'\0'						
-----	-----	-----	-----	-----	-----	------	--	--	--	--	--	--



رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

در این روش، برای نگهداری یک رشته، از یک آرایه از نوع `char` استفاده می‌کنیم (هر کاراکتر رشته در یک خانه از آرایه ذخیره می‌شود).

البته در حقیقت، به جای هر کاراکتر، کد اسکی آن نگهداری می‌شود.

ممکن است فضای آرایه، بیشتر از طول رشته باشد، لذا طبق قرارداد، در انتهای رشته، کاراکتر `'\0'` را قرار می‌دهیم.

دقت کنید که ثابت‌های رشته‌ای با "محدود می‌شدند ولی برای کاراکترها، هر کاراکتر بین دو ' قرار می‌گیرد، مثلاً کاراکتر `a` به صورت `'a'` بیان می‌شود.

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام `str` به صورت زیر عمل می‌کنیم:

```
char str[10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

str	'H'	'e'	'l'	'l'	'o'	'\0'						
-----	-----	-----	-----	-----	-----	------	--	--	--	--	--	--



رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

- در این روش، برای نگهداری یک رشته، از یک آرایه از نوع char استفاده می‌کنیم (هر کاراکتر رشته در یک خانه از آرایه ذخیره می‌شود).
 - البته در حقیقت، به جای هر کاراکتر، کد اسکی آن نگهداری می‌شود.
 - ممکن است فضای آرایه، بیشتر از طول رشته باشد، لذا طبق قرارداد، در انتهای رشته، کاراکتر '\0' را قرار می‌دهیم.
 - دقت کنید که ثابت‌های رشته‌ای با "محدود می‌شدند ولی برای کاراکترها، هر کاراکتر بین دو ' قرار می‌گیرد، مثلاً کاراکتر a به صورت 'a' بیان می‌شود.
 - مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام str به صورت زیر عمل می‌کنیم:
- ```
char str[10]={'H','e','l','l','o','\0'};
```
- به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

|     |     |     |     |     |     |      |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|
| str | 'H' | 'e' | 'l' | 'l' | 'o' | '\0' |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|



## رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام str به صورت زیر عمل می‌کنیم:

```
char str [10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

|     |     |     |     |     |     |      |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|
| str | 'H' | 'e' | 'l' | 'l' | 'o' | '\0' |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|

دقت کنید که پایان هر رشته با کاراکتر '\0' مشخص می‌شود.

لذا اولاً باید یک کاراکتر بیشتر از طول رشته فضا برای این کاراکتر در نظر گرفت.

ثانیاً اگر به هر دلیلی این کاراکتر حذف شود، در پردازش رشته مشکل ایجاد شده و گاهی کاراکترهای بی‌معنی به انتهای رشته اضافه می‌کند.



## رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام str به صورت زیر عمل می‌کنیم:

```
char str [10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

|     |     |     |     |     |     |      |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|
| str | 'H' | 'e' | 'l' | 'l' | 'o' | '\0' |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|

دقت کنید که پایان هر رشته با کاراکتر '\0' مشخص می‌شود.

لذا اولاً باید یک کاراکتر بیشتر از طول رشته فضا برای این کاراکتر در نظر گرفت.

ثانیاً اگر به هر دلیلی این کاراکتر حذف شود، در پردازش رشته مشکل ایجاد شده و گاهی کاراکترهای بی‌معنی به انتهای رشته اضافه می‌کند.



## رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام str به صورت زیر عمل می‌کنیم:

```
char str [10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

|     |     |     |     |     |     |      |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|
| str | 'H' | 'e' | 'l' | 'l' | 'o' | '\0' |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|

دقت کنید که پایان هر رشته با کاراکتر '\0' مشخص می‌شود.

لذا اولاً باید یک کاراکتر بیشتر از طول رشته فضا برای این کاراکتر در نظر گرفت.

ثانیاً اگر به هر دلیلی این کاراکتر حذف شود، در پردازش رشته مشکل ایجاد شده و گاهی کاراکترهای بی‌معنی به انتهای رشته اضافه می‌کند.



## رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام str به صورت زیر عمل می‌کنیم:

```
char str [10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

|     |     |     |     |     |     |      |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|
| str | 'H' | 'e' | 'l' | 'l' | 'o' | '\0' |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|

دقت کنید که پایان هر رشته با کاراکتر '\0' مشخص می‌شود.

لذا اولاً باید یک کاراکتر بیشتر از طول رشته فضا برای این کاراکتر در نظر گرفت.

ثانیاً اگر به هر دلیلی این کاراکتر حذف شود، در پردازش رشته مشکل ایجاد شده و گاهی کاراکترهای بی معنی به انتهای رشته اضافه می‌کند.



## رشته‌ها: آرایه‌ای از کاراکترها

رشته‌ها: آرایه‌ای از کاراکترها

مثلاً برای نگهداری رشته "Hello" در یک متغیر به نام str به صورت زیر عمل می‌کنیم:

```
char str [10]={'H','e','l','l','o','\0'};
```

به عبارت دقیق‌تر، کاراکترها به صورت زیر در آرایه قرار می‌گیرند.

|     |     |     |     |     |     |      |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|
| str | 'H' | 'e' | 'l' | 'l' | 'o' | '\0' |  |  |  |  |  |
|-----|-----|-----|-----|-----|-----|------|--|--|--|--|--|

دقت کنید که پایان هر رشته با کاراکتر '\0' مشخص می‌شود.

لذا اولاً باید یک کاراکتر بیشتر از طول رشته فضا برای این کاراکتر در نظر گرفت.

ثانیاً اگر به هر دلیلی این کاراکتر حذف شود، در پردازش رشته مشکل ایجاد شده و گاهی کاراکترهای بی معنی به انتهای رشته اضافه می‌کند.



# رشته‌ها: آرایه‌ای از کاراکترها

## ورودی و خروجی رشته‌ها

- برای خروجی رشته‌ها می‌توان مشابه سایر متغیرها عمل کرد، مثلاً `cout<<str;`
- اما برای ورودی، از دستورات `cin.get( str ,n);` یا `cin.getline ( str ,n);` استفاده می‌شود.
- در دستور فوق، `str` آرایه کاراکتری است که برای نگهداری رشته قبلاً تعریف شده است و `n` حداکثر طول رشته دریافتی است.
- معمولاً این مقدار یکی از فضای در نظر گرفته شده برای رشته کمتر است. دقت کنید که این یک کاراکتر را برای کاراکتر انتهای رشته نیاز داشتیم.
- در مواردی که رشته از ورودی گرفته می‌شود، کاراکتر انتهای رشته به صورت خودکار در پایان رشته اضافه می‌شود.
- البته علاوه بر این دستورات، دستورات دیگری نیز موجود است که در آینده به آن خواهیم پرداخت.



# رشته‌ها: آرایه‌ای از کاراکترها

## ورودی و خروجی رشته‌ها

- برای خروجی رشته‌ها می‌توان مشابه سایر متغیرها عمل کرد، مثلاً `cout<<str;`
- اما برای ورودی، از دستورات `cin.get(str, n);` یا `cin.getline(str, n);` استفاده می‌شود.
- در دستور فوق، `str` آرایه کاراکتری است که برای نگهداری رشته قبلاً تعریف شده است و `n` حداکثر طول رشته دریافتی است.
- معمولاً این مقدار یکی از فضای در نظر گرفته شده برای رشته کمتر است. دقت کنید که این یک کاراکتر را برای کاراکتر انتهای رشته نیاز داشتیم.
- در مواردی که رشته از ورودی گرفته می‌شود، کاراکتر انتهای رشته به صورت خودکار در پایان رشته اضافه می‌شود.
- البته علاوه بر این دستورات، دستورات دیگری نیز موجود است که در آینده به آن خواهیم پرداخت.



# رشته‌ها: آرایه‌ای از کاراکترها

## ورودی و خروجی رشته‌ها

- برای خروجی رشته‌ها می‌توان مشابه سایر متغیرها عمل کرد، مثلاً `cout<<str;`
- اما برای ورودی، از دستورات `cin.get(str, n);` یا `cin.getline(str, n);` استفاده می‌شود.
- در دستور فوق، `str` آرایه کاراکتری است که برای نگهداری رشته قبلاً تعریف شده است و `n` حداکثر طول رشته دریافتی است.
- معمولاً این مقدار یکی از فضای در نظر گرفته شده برای رشته کمتر است. دقت کنید که این یک کاراکتر را برای کاراکتر انتهای رشته نیاز داشتیم.
- در مواردی که رشته از ورودی گرفته می‌شود، کاراکتر انتهای رشته به صورت خودکار در پایان رشته اضافه می‌شود.
- البته علاوه بر این دستورات، دستورات دیگری نیز موجود است که در آینده به آن خواهیم پرداخت.



# رشته‌ها: آرایه‌ای از کاراکترها

## ورودی و خروجی رشته‌ها

- برای خروجی رشته‌ها می‌توان مشابه سایر متغیرها عمل کرد، مثلاً `cout<<str;`
- اما برای ورودی، از دستورات `cin.get(str, n);` یا `cin.getline(str, n);` استفاده می‌شود.
- در دستور فوق، `str` آرایه کاراکتری است که برای نگهداری رشته قبلاً تعریف شده است و `n` حداکثر طول رشته دریافتی است.
- معمولاً این مقدار یکی از فضای در نظر گرفته شده برای رشته کمتر است. دقت کنید که این یک کاراکتر را برای کاراکتر انتهای رشته نیاز داشتیم.
- در مواردی که رشته از ورودی گرفته می‌شود، کاراکتر انتهای رشته به صورت خودکار در پایان رشته اضافه می‌شود.
- البته علاوه بر این دستورات، دستورات دیگری نیز موجود است که در آینده به آن خواهیم پرداخت.



# رشته‌ها: آرایه‌ای از کاراکترها

## ورودی و خروجی رشته‌ها

- برای خروجی رشته‌ها می‌توان مشابه سایر متغیرها عمل کرد، مثلاً `cout<<str;`
- اما برای ورودی، از دستورات `cin.get(str, n);` یا `cin.getline(str, n);` استفاده می‌شود.
- در دستور فوق، `str` آرایه کاراکتری است که برای نگهداری رشته قبلاً تعریف شده است و `n` حداکثر طول رشته دریافتی است.
- معمولاً این مقدار یکی از فضای در نظر گرفته شده برای رشته کمتر است. دقت کنید که این یک کاراکتر را برای کاراکتر انتهای رشته نیاز داشتیم.
- در مواردی که رشته از ورودی گرفته می‌شود، کاراکتر انتهای رشته به صورت خودکار در پایان رشته اضافه می‌شود.
- البته علاوه بر این دستورات، دستورات دیگری نیز موجود است که در آینده به آن خواهیم پرداخت.



# رشته‌ها: آرایه‌ای از کاراکترها

## ورودی و خروجی رشته‌ها

- برای خروجی رشته‌ها می‌توان مشابه سایر متغیرها عمل کرد، مثلاً `cout<<str;`
- اما برای ورودی، از دستورات `cin.get(str, n);` یا `cin.getline(str, n);` استفاده می‌شود.
- در دستور فوق، `str` آرایه کاراکتری است که برای نگهداری رشته قبلاً تعریف شده است و `n` حداکثر طول رشته دریافتی است.
- معمولاً این مقدار یکی از فضای در نظر گرفته شده برای رشته کمتر است. دقت کنید که این یک کاراکتر را برای کاراکتر انتهای رشته نیاز داشتیم.
- در مواردی که رشته از ورودی گرفته می‌شود، کاراکتر انتهای رشته به صورت خودکار در پایان رشته اضافه می‌شود.
- البته علاوه بر این دستورات، دستورات دیگری نیز موجود است که در آینده به آن خواهیم پرداخت.



## مثال ۱

برنامه‌ای بنویسید که یک رشته را از ورودی دریافت کند و طول رشته (تعداد کاراکترهای آن) و تعداد کاراکترهای  $a$  را در آن محاسبه و چاپ کند.



## مثال ۱

برنامه‌ای بنویسید که یک رشته را از ورودی دریافت کند و طول رشته (تعداد کاراکترهای آن) و تعداد کاراکترهای a را در آن محاسبه و چاپ کند.

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5 char str[101];
6 int i, counter=0;
7 cout<<"Enter String :";
8 cin.get(str,100);
9 for(i=0; str[i] != '\0'; i++)
10 if (str[i] == 'a')
11 counter++;
12 cout<<"String Length="<<i<<"Number of a's:"<<counter;
13 return 0;
14 }
```



## مثال ۲

برنامه‌ای بنویسید که یک رشته و دو کاراکتر را از ورودی دریافت کند و در رشته، کاراکتر اول را به کاراکتر دوم تبدیل کرده و رشته حاصل را چاپ کند.



## مثال ۲

برنامه‌ای بنویسید که یک رشته و دو کاراکتر را از ورودی دریافت کند و در رشته، کاراکتر اول را به کاراکتر دوم تبدیل کرده و رشته حاصل را چاپ کند.

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5 char str[101], ch1, ch2;
6 int i;
7 cout<<"Enter String :";
8 cin.get(str,100);
9 cout<<"Enter First Character:";
10 cin>>ch1;
11 cout<<"Enter Second Character:";
12 cin>>ch2;
13 for(i=0; str[i] != '\0'; i++)
14 if (str[i] == ch1)
15 str[i]=ch2;
16 cout<<"Result="<<str;
17 return 0;
18 }
```



دانشگاه یزد  
فakولت علوم ریاضی

# کتابخانهٔ string.h یا cstring

## کتابخانهٔ string.h یا cstring

برای سادگی کار با رشته‌ها، کتابخانه‌های متعددی وجود دارد.

یکی از این کتابخانه‌ها که در زبان C ارائه شد، کتابخانهٔ string.h است.

برای استفاده از این کتابخانه، کافی است یکی از دو دستور زیر را در ابتدای برنامهٔ خود قرار دهید:

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

```
#include <cstring>
```

دقت کنید که این کتابخانه، با کتابخانه string که در زبان C++ ارائه شده است متفاوت است.

تفاوت این دو در استفاده h. در انتهای اولی است. در C++، این کلاس با cstring نیز نامیده می‌شود

و می‌توان به جای string.h از cstring نیز در دستورات استفاده کرد.



# کتابخانهٔ string.h یا cstring

## کتابخانهٔ string.h یا cstring

برای سادگی کار با رشته‌ها، کتابخانه‌های متعددی وجود دارد.

یکی از این کتابخانه‌ها که در زبان C ارائه شد، کتابخانهٔ string.h است.

برای استفاده از این کتابخانه، کافی است یکی از دو دستور زیر را در ابتدای برنامهٔ خود قرار دهید:

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

```
#include <cstring>
```

دقت کنید که این کتابخانه، با کتابخانه string که در زبان C++ ارائه شده است متفاوت است.

تفاوت این دو در استفاده h. در انتهای اولی است. در C++، این کلاس با cstring نیز نامیده می‌شود و می‌توان به جای string.h از cstring نیز در دستورات استفاده کرد.



# کتابخانهٔ string.h یا cstring

## کتابخانهٔ string.h یا cstring

برای سادگی کار با رشته‌ها، کتابخانه‌های متعددی وجود دارد.

یکی از این کتابخانه‌ها که در زبان C ارائه شد، کتابخانهٔ string.h است.

برای استفاده از این کتابخانه، کافی است یکی از دو دستور زیر را در ابتدای برنامهٔ خود قرار دهید:

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

```
#include <cstring>
```

دقت کنید که این کتابخانه، با کتابخانه string که در زبان C++ ارائه شده است متفاوت است.

تفاوت این دو در استفاده h. در انتهای اولی است. در C++، این کلاس با cstring نیز نامیده می‌شود و می‌توان به جای string.h از cstring نیز در دستورات استفاده کرد.



# کتابخانهٔ string.h یا cstring

## کتابخانهٔ string.h یا cstring

برای سادگی کار با رشته‌ها، کتابخانه‌های متعددی وجود دارد.

یکی از این کتابخانه‌ها که در زبان C ارائه شد، کتابخانهٔ string.h است.

برای استفاده از این کتابخانه، کافی است یکی از دو دستور زیر را در ابتدای برنامهٔ خود قرار دهید:

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

```
#include <cstring>
```

دقت کنید که این کتابخانه، با کتابخانه string که در زبان C++ ارائه شده است متفاوت است.

تفاوت این دو در استفاده h. در انتهای اولی است. در C++، این کلاس با cstring نیز نامیده می‌شود و می‌توان به جای string.h از cstring نیز در دستورات استفاده کرد.



# کتابخانهٔ string.h یا cstring

## کتابخانهٔ string.h یا cstring

برای سادگی کار با رشته‌ها، کتابخانه‌های متعددی وجود دارد.

یکی از این کتابخانه‌ها که در زبان C ارائه شد، کتابخانهٔ string.h است.

برای استفاده از این کتابخانه، کافی است یکی از دو دستور زیر را در ابتدای برنامهٔ خود قرار دهید:

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

```
#include <cstring>
```

دقت کنید که این کتابخانه، با کتابخانه string که در زبان C++ ارائه شده است متفاوت است.

تفاوت این دو در استفاده h. در انتهای اولی است. در C++، این کلاس با cstring نیز نامیده می‌شود و می‌توان به جای string.h از cstring نیز در دستورات استفاده کرد.



# کتابخانهٔ string.h یا cstring

برخی از توابع این کتابخانه به صورت زیر است:

تابع `strcpy`: این تابع دو رشته را به عنوان پارامتر گرفته و پارامتر دوم را در پارامتر اول کپی می‌کند. در حقیقت این تابع نقش عمل انتساب برای رشته‌ها را بازی می‌کند. شکل استفاده از این تابع به صورت زیر است:

```
strcpy(char destination[], char source[])
```

تابع `strlen`: این تابع یک رشته را به عنوان پارامتر گرفته و طول رشته، یعنی تعداد کاراکترهای آن را برمی‌گرداند. شکل استفاده از این تابع به صورت زیر است:

```
strlen(char str[])
```

تابع `strcmp`: این تابع دو رشته را به عنوان پارامتر گرفته و آن دو را به یکدیگر مقایسه می‌کند. مقایسه به صورت لغتنامه‌ای است. اگر از نظر لغتنامه‌ای رشته اول، قبل از رشته دوم قرار گیرد، که در این حالت گوییم رشته اول کوچک‌تر از رشته دوم است، مقدار منفی، در صورت تساوی دو رشته، مقدار صفر و در صورت بزرگ‌تر بودن رشته اول، مقدار مثبت توسط این تابع برگشت داده می‌شود.

```
strcmp(char str1[], char str2[])
```



# کتابخانهٔ string.h یا cstring

برخی از توابع این کتابخانه به صورت زیر است:

تابع `strcpy`: این تابع دو رشته را به عنوان پارامتر گرفته و پارامتر دوم را در پارامتر اول کپی می‌کند. در حقیقت این تابع نقش عمل انتساب برای رشته‌ها را بازی می‌کند. شکل استفاده از این تابع به صورت زیر است:

```
strcpy(char destination[], char source[])
```

تابع `strlen`: این تابع یک رشته را به عنوان پارامتر گرفته و طول رشته، یعنی تعداد کاراکترهای آن را برمی‌گرداند. شکل استفاده از این تابع به صورت زیر است:

```
strlen(char str[])
```

تابع `strcmp`: این تابع دو رشته را به عنوان پارامتر گرفته و آن دو را به یکدیگر مقایسه می‌کند. مقایسه به صورت لغتنامه‌ای است. اگر از نظر لغتنامه‌ای رشته اول، قبل از رشته دوم قرار گیرد، که در این حالت گوییم رشته اول کوچک‌تر از رشته دوم است، مقدار منفی، در صورت تساوی دو رشته، مقدار صفر و در صورت بزرگ‌تر بودن رشته اول، مقدار مثبت توسط این تابع برگشت داده می‌شود.

```
strcmp(char str1[], char str2[])
```



# کتابخانهٔ string.h یا cstring

برخی از توابع این کتابخانه به صورت زیر است:

تابع `strcpy`: این تابع دو رشته را به عنوان پارامتر گرفته و پارامتر دوم را در پارامتر اول کپی می‌کند. در حقیقت این تابع نقش عمل انتساب برای رشته‌ها را بازی می‌کند. شکل استفاده از این تابع به صورت زیر است:

```
strcpy(char destination[], char source[])
```

تابع `strlen`: این تابع یک رشته را به عنوان پارامتر گرفته و طول رشته، یعنی تعداد کاراکترهای آن را برمی‌گرداند. شکل استفاده از این تابع به صورت زیر است:

```
strlen(char str[])
```

تابع `strcmp`: این تابع دو رشته را به عنوان پارامتر گرفته و آن دو را به یکدیگر مقایسه می‌کند. مقایسه به صورت لغتنامه‌ای است. اگر از نظر لغتنامه‌ای رشته اول، قبل از رشته دوم قرار گیرد، که در این حالت گوییم رشته اول کوچک‌تر از رشته دوم است، مقدار منفی، در صورت تساوی دو رشته، مقدار صفر و در صورت بزرگ‌تر بودن رشته اول، مقدار مثبت توسط این تابع برگشت داده می‌شود.

```
strcmp(char str1[], char str2[])
```



# کتابخانهٔ string.h یا cstring

برخی از توابع این کتابخانه به صورت زیر است:

تابع `strcat`: این تابع دو رشته را به عنوان پارامتر گرفته و پارامتر دوم را به انتهای پارامتر اول می‌پسباند. شکل استفاده از این تابع به صورت زیر است:

```
strcat(char str1[], char str2[])
```

توابع دیگری نیز در این کتابخانه وجود دارد که فوآننده علاقه‌مند می‌تواند از طریق وبسایت‌های مختلف یا کتب دیگر به اطلاعات آنها دسترسی پیدا کند.

این توابع، به سادگی قابلیت این را دارند که توسط فوآننده نوشته شوند. به عنوان تمرین می‌توانید تعدادی از این توابع را خودتان پیاده‌سازی نمایید.



# کتابخانهٔ string.h یا cstring

برخی از توابع این کتابخانه به صورت زیر است:

تابع `strcat`: این تابع دو رشته را به عنوان پارامتر گرفته و پارامتر دوم را به انتهای پارامتر اول می‌پسباند. شکل استفاده از این تابع به صورت زیر است:

```
strcat(char str1[], char str2[])
```

توابع دیگری نیز در این کتابخانه وجود دارد که فوآننده علاقه‌مند می‌تواند از طریق وبسایت‌های مختلف یا کتب دیگر به اطلاعات آنها دسترسی پیدا کند.

این توابع، به سادگی قابلیت این را دارند که توسط فوآننده نوشته شوند. به عنوان تمرین می‌توانید تعدادی از این توابع را خودتان پیاده‌سازی نمایید.



# کتابخانهٔ string.h یا cstring

برخی از توابع این کتابخانه به صورت زیر است:

تابع `strcat`: این تابع دو رشته را به عنوان پارامتر گرفته و پارامتر دوم را به انتهای پارامتر اول می‌پسباند. شکل استفاده از این تابع به صورت زیر است:

```
strcat(char str1[], char str2[])
```

توابع دیگری نیز در این کتابخانه وجود دارد که خواننده علاقه‌مند می‌تواند از طریق وبسایت‌های مختلف یا کتب دیگر به اطلاعات آنها دسترسی پیدا کند.

این توابع، به سادگی قابلیت این را دارند که توسط خواننده نوشته شوند. به عنوان تمرین می‌توانید تعدادی از این توابع را خودتان پیاده‌سازی نمایید.



## مثال ۳

برنامه‌ای بنویسید که ابتدا عدد  $n$  و سپس  $n$  نام را از ورودی دریافت کند و اسامی را به ترتیب صعودی مرتب کرده و چاپ کند.

```

1 #include <iostream>
2 using namespace std;
3 void Sort(double A[500], unsigned int n)
4 {
5 unsigned int i, j;
6 double B;
7 for (i=0; i<n; i++)
8 for (j=i+1; j<n; j++)
9 if (A[i]>A[j])
10 {
11 B=A[i];
12 A[i]=A[j];
13 A[j]=B;
14 }
15 return;
16 }
```



## مثال ۳

برنامه‌ای بنویسید که ابتدا عدد  $n$  و سپس  $n$  نام را از ورودی دریافت کند و اسامی را به ترتیب صعودی مرتب کرده و چاپ کند.

```

1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 void Sort(char A[100][31], unsigned int n)
5 {
6 unsigned int i, j;
7 char B[31];
8 for(i=0; i<n; i++)
9 for(j=i+1; j<n; j++)
10 if (strcmp(A[i], A[j]) > 0)
11 {
12 strcpy(B, A[i]);
13 strcpy(A[i], A[j]);
14 strcpy(A[j], B);
15 }
16 }
```



```

1 #include <iostream>
2 using namespace std;
3 void Sort(double A[500], unsigned int n)
4 {
5 unsigned int i, j;
6 double B;
7 for(i=0; i<n; i++)
8 for(j=i+1; j<n; j++)
9 if (A[i]>A[j])
10 {
11 B=A[i];
12 A[i]=A[j];
13 A[j]=B;
14 }
15 return;
16 }
```

## مثال ۴

برنامه‌ای بنویسید که ابتدا عدد  $n$  و سپس  $n$  نام را از ورودی دریافت کند و اسامی را به ترتیب صعودی مرتب کرده و چاپ کند.

```

1 void Input_Array(double A[500], int n)
2 {
3 int i;
4 for(i=0; i<n; i++)
5 {
6 cout<<"Enter Next Element:";
7 cin>>A[i];
8 }
9 return;
10 }
11 void Output_Array(double A[500], int n)
12 {
13 int i;
14 for(i=0; i<n; i++)
15 cout<<A[i]<<" ";
16 return;
17 }
```



## مثال ۴

برنامه‌ای بنویسید که ابتدا عدد  $n$  و سپس  $n$  نام را از ورودی دریافت کند و اسامی را به ترتیب صعودی مرتب کرده و چاپ کند.

```

1 void Input_Strings(char A[100][31], int n)
2 {
3 int i;
4 for(i=0; i<n; i++)
5 {
6 cin.ignore();
7 cout<<"Enter Next String:";
8 cin.get(A[i],30);
9 }
10 }
11 void Output_Strings(char A[100][31], int n)
12 {
13 int i;
14 for(i=0; i<n; i++)
15 cout<<A[i]<<"\n";
16 }

```



```

1 void Input_Array(double A[500], int n)
2 {
3 int i;
4 for(i=0; i<n; i++)
5 {
6 cout<<"Enter Next Element:";
7 cin>>A[i];
8 }
9 return;
10 }
11 void Output_Array(double A[500], int n)
12 {
13 int i;
14 for(i=0; i<n; i++)
15 cout<<A[i]<<" ";
16 return;
17 }

```

## مثال ۵

برنامه‌ای بنویسید که ابتدا عدد  $n$  و سپس  $n$  نام را از ورودی دریافت کند و اسامی را به ترتیب صعودی مرتب کرده و چاپ کند.

```

1 int main()
2 {
3 unsigned int n;
4 double A[500];
5 cout << "Enter n: ";
6 cin >> n;
7 Input_Array(A,n);
8 Sort(A,n);
9 Output_Array(A,n);
10 return 0;
11 }
```



## مثال ۵

برنامه‌ای بنویسید که ابتدا عدد  $n$  و سپس  $n$  نام را از ورودی دریافت کند و اسامی را به ترتیب صعودی مرتب کرده و چاپ کند.

```

1 int main()
2 {
3 unsigned int n;
4 char A[100][31];
5 cout << "Enter n: ";
6 cin >> n;
7 Input_Strings(A,n);
8 Sort(A,n);
9 Output_Strings(A,n);
10 return 0;
11 }
```

```

1 int main()
2 {
3 unsigned int n;
4 double A[500];
5 cout << "Enter n: ";
6 cin >> n;
7 Input_Array(A,n);
8 Sort(A,n);
9 Output_Array(A,n);
10 return 0;
11 }
```



## مثال ۶

برنامه‌ای بنویسید که یک رشته را از ورودی دریافت کرده و آن را به دو رشته به صورت زیر تفکیک کرده و ماصل را چاپ کند. کاراکترهای زوج رشته را در رشته اول و کاراکترهای فرد آن را در رشته دوم قرار دهد.



```

1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 void Split_String (char Source[], char Odd[], char Even[])
5 {
6 int i;
7 for (i=0; i<= strlen(Source); i++)
8 if (i%2 == 1)
9 Odd[i/2]=Source[i];
10 else
11 Even[i/2]=Source[i];
12 Odd[i/2]=Even[i /2]='\0';
13 }
14 int main()
15 {
16 unsigned int n;
17 char A[101], B[51], C[51];
18 cout << "Enter an String: ";
19 cin.get(A,100);
20 Split_String (A,B,C);
21 cout<<"\nOdd String="<<B<<"\nEven String="<<C;
22 return 0;
23 }

```

### مثال ۶

برنامه‌ای بنویسید که یک رشته را از ورودی دریافت کرده و آن را به دو رشته به صورت زیر تفکیک کرده و حاصل را چاپ کند. کاراکترهای زوج رشته را در رشته اول و کاراکترهای فرد آن را در رشته دوم قرار دهد.

با آرزوی موفقیت  
و بهروزی