



به نام خدا
دانشکده مهندسی برق و کامپیوتر دانشکده فنی
دانشگاه تهران
مبانی کامپیوتر و برنامه‌نویسی



استاد : دکتر مرادی،
دکتر هاشمی

عنوان:
آرایه‌ها

نیمسال اول
۰۱-۰۲

در این جلسه شما با آرایه^۱ ها و نوع خاصی از آن ها یعنی رشته^۲ ها آشنا خواهید شد.

تعریف آرایه: تعدادی داده های پشت سر هم در حافظه که همگی از یک نوع^۳ هستند.

Index number		Memory address
x[0]	7	2293396
x[1]	62	2293490
x[2]	-12	2293404
x[3]	256	2293400
x[4]	-116	2293412
x[5]	10	2293416

Array name Data stored in memory

نکته ۱: شماره اندیس آرایه به طول n از 0 شروع و تا n-1 ادامه دارد.

برای تعریف و مقدار دهی آرایه‌ها در زبان C روش های متعددی وجود دارد. در کد زیر تعدادی از این روش ها ذکر شده‌اند.

```
✓ int a[] = { 5, 3, 2 };  
✓ int b[3];  
✓ int c[3] = { 6, 5 };  
✓ b[0] = 2;  
  
✓ int d[3];  
✗ d = { 4, 5, 6 };  
✓ int e[3];  
✗ e[3] = 4;
```

نکته ۲: مقداردهی آرایه با استفاده از {} تنها هنگام تعریف آرایه مجاز است.

نکته ۳: طول آرایه‌ها باید هنگام در زمان compile تعیین شود. بدین معنا که طول آرایه در زمان اجرای برنامه نمی توان تعریف یا تغییر داده شود. با توجه به این موضوع طول آرایه باید عدد ثابتی داشته باشد. میتوان با دستور #define مقدار ثابتی را تعریف نمود.

¹ array
² string
³ type

← ۱. انجام دهید!

۱. یک آرایه از نوع `int` به طول ۵ تعریف کنید. طول آرایه را با استفاده از یک متغیر تعریف کنید. برنامه خود را کامپایل کرده و نتیجه را ببینید. چه خطایی رخ میدهد و چرا؟
۲. برنامه بالا را با تغییر متغیر به یک متغیر `const` تغییر داده و مجدداً کامپایل کنید. آیا خطا رخ نمی‌دهد؟ چرا؟
۳. با استفاده از حلقه `for` و دستور `scanf`، ۵ عدد از کاربر دریافت نموده و در خانه‌های آرایه ذخیره کنید.
۴. با استفاده از دستور `printf` مقادیر وارد شده را به ترتیب معکوس در خروجی چاپ کنید.

راهنمایی:

```
scanf("%d", &a[i]);
```

قسمت ۱: نتیجه را به دستیاران آموزشی نشان دهید.

← ۲. انجام دهید!

- ۱) در تابع `main` یک آرایه از کاراکترها (به طول ۵) به نام `msg` بسازید و سعی کنید به عنوان مقدار اولیه رشته ی "Hello" را در آن بریزید.
- ۲) به `compile error` تولید شده دقت کنید. چرا با وجود آن که طول رشته و طول آرایه برابر است این خطا تولید شده است؟ درباره `NULL character` یا `'\0'` تحقیق کنید.
- ۳) حال به وسیله ی یک حلقه `for` طول رشته ی `msg` را محاسبه و بر روی صفحه چاپ کنید.

قسمت ۲: نتیجه را به دستیاران آموزشی نشان دهید.

← ۳. انجام دهید!

یک آرایه را نمی‌توان برابر یک آرایه ی دیگر گذاشت. مثلاً برنامه ی زیر خطای کامپایلی دارد.

```
int arr[4] = {1, 2, 3, 4};  
int arr_copy[4];  
arr_copy = arr;
```

۱. برنامه را کامپایل کرده و خطا را بنویسید.
۲. چرا این خطا رخ میدهد؟
۳. کد را بگونه ای تغییر دهید تا مشکل حل شده و محتوای یک آرایه را در یک آرایه دیگر ذخیره کنید.

قسمت ۳: نتیجه را به دستیاران آموزشی نشان دهید.

۴. انجام دهید

در این قسمت، به عنوان ورودی برنامه، آرایه‌ای از اعداد طبیعی به شما داده می‌شود و خواسته‌ی مسئله این است که برای هر یک از اعداد موجود در آرایه، اولین عددی را که در سمت چپ آن بوده و بزرگتر از آن است، بیابید. چنانچه چنین عددی وجود نداشت، مقدار پیش‌فرض صفر برای آن در نظر گرفته می‌شود. به عنوان مثال، رشته‌ی زیر را در نظر بگیرید:

{3, 2, 5, 4, 1, 2, 3, 8, 3, 6}

اگر از چپ به راست روی عناصر این رشته حرکت کنیم، به عنوان مثال برای عنصر سوم که عدد ۵ است، پاسخ ۱- خواهد بود. یا برای عنصر شماره‌ی ۳ که عدد ۴ است، پاسخ اندیس مربوط به عدد ۵، یعنی ۲ است. چنانچه در زیر مشاهده می‌شود، پاسخ کلی مثال بالا این است:

{-1, 0, -1, 2, 3, 3, 3, -1, 7, 7}

حال مراحل زیر را به ترتیب انجام دهید:

- ۱) یک آرایه از اعداد صحیح و به طول ۱۰، با مقادیر دلخواه خودتان ایجاد کنید. (می‌توانید با استفاده از تولید کردن اعداد تصادفی این کار را بکنید. در آزمایش‌های قبلی این کار را یاد گرفته‌اید:)
- ۲) ابتدا مقدار میانگین اعداد آرایه را محاسبه و چاپ نمایید. (برای این کار با استفاده از حلقه **for** جمع اعداد آرایه را حساب کرده و بر ۱۰ تقسیم کنید. برای دقیق‌تر بودن پاسخ از متغیر **float** برای نگه داری جمع اعداد استفاده کنید.)
- ۳) اکنون، الگوریتمی را پیدا کنید که در آن با استفاده از حلقه **for**، بتوان برای هر یک از اعداد موجود در آرایه، اولین عددی را که در اندیس‌های کمتر از آن قرار داشته و از آن بزرگتر است، یافت. به نظر شما این روش به ازای **n** تا ورودی، ماکزیمم چند بار پیمایش روی آرایه خواهد داشت؟ آیا راه سریعتری وجود دارد؟

قسمت 4: نتیجه را به دستیاران آموزشی نشان دهید.

آرایه های چند بعدی:

در زبان **C** می‌توان آرایه‌هایی با بیش از یک بعد نیز تعریف و استفاده کرد. نحوه‌ی تعریف یک آرایه از نوع کاراکتر با **n** بعد به صورت زیر است:

تا **n** براکت

char x [8][10]...[6];

↑ طول بعد اول

↑ طول بعد دوم

....

↑ طول بعد **n**ام

دقت کنید: حافظه ی کامپیوتر مانند یک آرایه ی یک بعدی است. لذا برای شبیه سازی آرایه هایی با ابعاد بیش تر سطرها ی آن را پشت سر هم قرار می دهد و با استفاده از اشاره گر به آن ها دسترسی پیدا می کند.

به عنوان مثال اگر آرایه از جنس `int` است:

```
int a[3][4]={11,12,14,15},{21,22,23,24}, {13,31,32,33};
```

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

همچنین می توانید به این شکل هم آرایه ی فوق را تعریف کنید:

```
int a[3][4]={11,12,14,15,21,22,23,24,13,31,32,33};
```

دقت کنید که همانند آرایه های یک بعدی شما فقط هنگام تعریف یک آرایه ی چند بعدی می توانید آن را به صورت فوق مقداردهی کنید.

۵. انجام دهید

برنامه ی زیر را کامل کنید. در حلقه ی `for` دوم می بایست قطعه کد کوچکی بنویسید که عناصر آرایه ی با نام `second_array` را در آرایه ی با نام `table` بریزد.

```
#include <stdio.h>
#define SIZE 5

int main() {
    int i, j;
    int table[SIZE][SIZE] = { { 1,2,3,4,5 }, { 6,7,8,9,10 }, { 11,12,13,14,15 }, {
1,2,3,4,5 }, { 4,6,7,3,2 } };
    int second_table[SIZE][SIZE];

    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            second_table[j][i] = table[i][j];
        }
    } // transposing

    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
```

```
        // Your code goes here
    }
} // replacing

for (i = 0; i < SIZE; i++) {
    for (j = 0; j < SIZE; j++)
        printf("%d ", table[i][j]);
    printf("\n");
} // printing the result

return 0;
}
```

قسمت 5: نتیجه را به دستیاران آموزشی نشان دهید.

بخش های اختیاری در یک فایل جداگانه بر روی سایت قرار گرفته است. توصیه می شود که حتماً آن ها را انجام دهید.

موفق باشید