



به نام خدا

# دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران

## مبانی کامپیوتر و برنامه نویسی



استاد: دکتر مرادی

عنوان: آزمایشگاه چهارم (آرایه ها)

نیمسال دوم

۹۸-۹۹

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

تعریف آرایه: تعدادی داده های پشت سر هم در حافظه که همگی از یک نوع<sup>۳</sup> 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 آرایه ها در زبان C روش های متعددی وجود دارد. در کد زیر تعدادی از این روش ها ذکر شده اند.

```
✓ 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: مقدار دهی آرایه با استفاده از {} تنها هنگام تعریف آرایه مجاز است.

نکته ۳: طول آرایه ها 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 می تواند به صورت متغیر const تعریف شود.

<sup>1</sup> array

<sup>2</sup> string

<sup>3</sup> type

## ۱. انجام دهید!

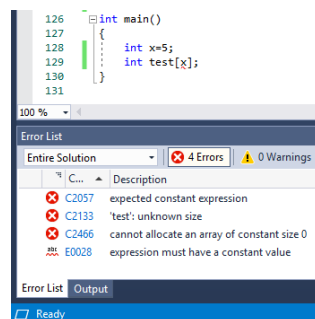
۱. یک آرایه از نوع `int` به طول ۵ تعریف کنید. طول آرایه را با استفاده از یک متغیر تعریف کنید. برنامه خود را کامپایل کرده و نتیجه را ببینید. چه خطایی رخ میدهد و چرا؟
۲. برنامه بالا را با تغییر متغیر به یک متغیر `const` تغییر داده و مجدداً کامپایل کنید. آیا خطا رخ نمی‌دهد؟ چرا؟
۳. با استفاده از حلقه `for` 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 :** موارد خواسته شده را انجام دهید. نتایج به دست آمده و یافته‌های خود را در کادر زیر بنویسید.

در بخش اول، ۴ خطا رخ میدهد که مضمون همه آنها این است که طول آرایه باید یک مقدار ثابت باشد.



در بخش دوم، اگر در `cpp` باشد، بدون مشکل کامپایل میکند. اما در `C` همان ارور ها تکرار میشود.

کد نوشته شده برای بخش سوم و چارم بصورت زیر میباشد :

```
int main()  
{  
    int test[5];  
    for (int i = 0; i < 5; i++)  
    {  
        scanf_s("%d", &test[i]);  
    }  
    for (int i = 0; i < 5 ; i++)  
    {  
        printf("%d\t", test[4 - i]);  
    }  
}
```

## ← 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:** موارد خواسته شده را انجام دهید. نتایج به دست آمده و یافته‌های خود را در کادر زیر بنویسید.

ما با خطایی مواجه میشویم که میگوید **overflow** رخ داده و نمیتوان رشته کاراکتر با طول ۶ را در یک آرایه با طول ۵ ریخت. اما رشته کاراکتری که ما تعریف کردیم ۵ حرفی بود. در واقع در زبان C در پایان این رشته کاراکتر '\0' میگذارد که پایان دهنده رشته کاراکتر میباشد و خود یک خانه از آرایه را اشغال میکند. پس برای اینکه ببینیم چند خانه از یک آرایه توسط رشته کاراکتری Hello ایجاد میشود میتوان از قطعه کد زیر استفاده کرد :

```
int main()
{
    int count = 0;
    char msg[] = { "Hello" };
    for ( int i = 0; msg[i] >= 0 ; i++)
    {
        printf("%c", msg[i]);
        count++;
    }
    printf("\n%d", count);
}
```

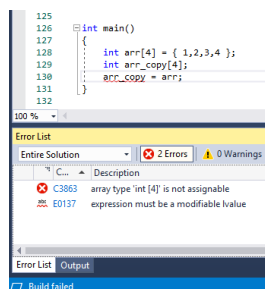
### ۳. انجام دهید! ←

یک آرایه را نمی توان 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:** موارد خواسته شده را انجام دهید. نتایج به دست آمده و یافته های خود را در کادر زیر بنویسید.



علت خطا این است که نمیتوان کل یک آرایه را برابر آرایه دیگر قرار داد زیرا یک آرایه نشانه ی یک آدرس است که در آن مقادیری ذخیره میشود. برای کپی کردن مقادیر یک آرایه در آرایه دیگر که طول برابر با اولی دارد میتوان از قطعه کد زیر استفاده کرد :

```
int main()
{
    int arr[4] = { 1,2,3,4 };
    int arr_copy[4];
    for (int i = 0; i < 4; i++)
    {
        arr_copy[i] = arr[i];
    }
}
```

#### ۴. انجام دهید!

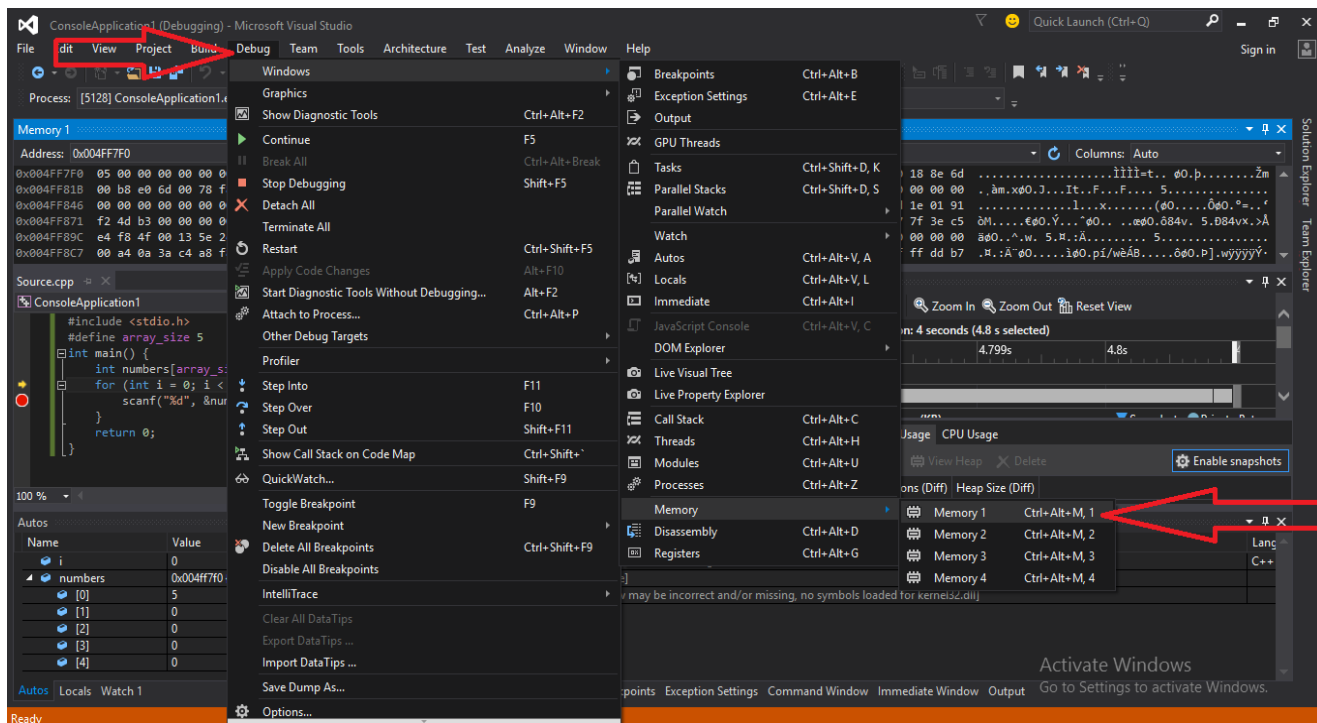
(۱) مشابه قطعه کد زیر، آرایه‌ای به طول ۵ تعریف کرده و یک break point بر روی دستور scanf تعریف کنید:

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

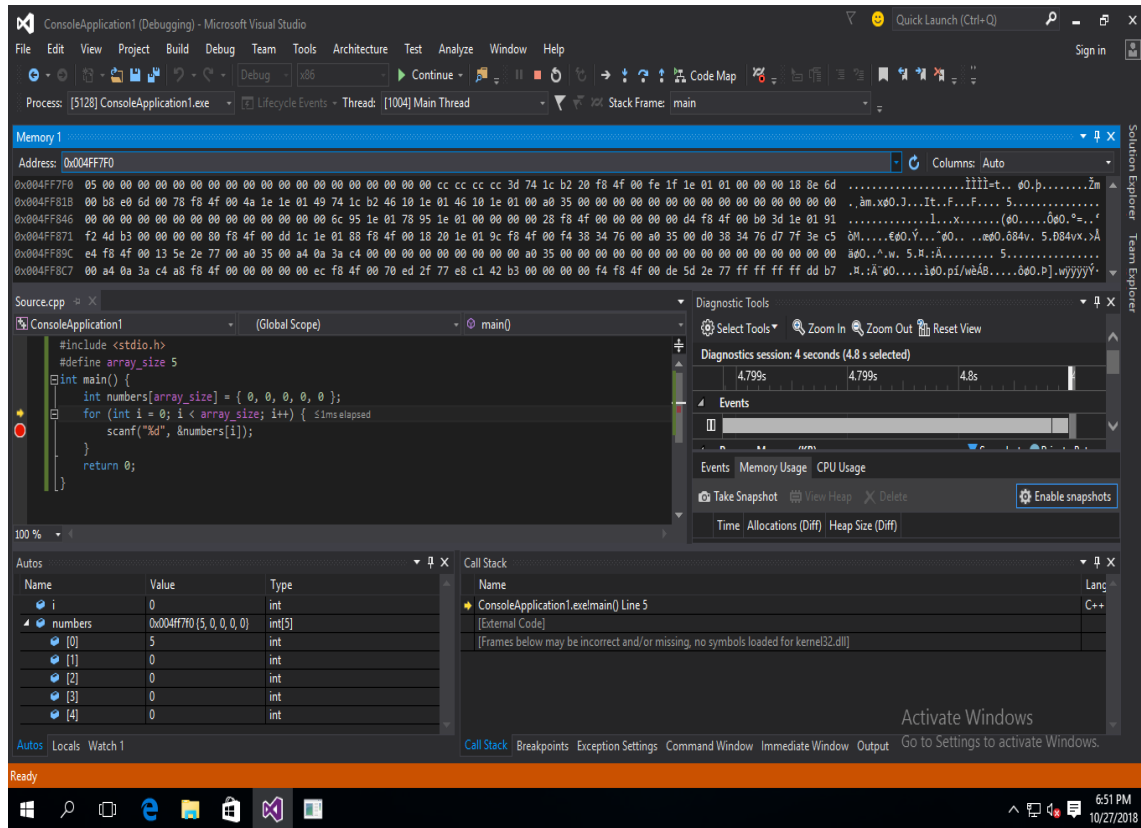
int main() {
    int numbers[array_size] = { 0, 0, 0, 0, 0 };
    for (int i = 0; i < array_size; i++) {
        scanf("%d", &numbers[i]);
    }
    return 0;
}
```

(۲) کد خود را با Debug mode اجرا کنید.

(۳) مطابق شکل زیر، پنجره‌ای که مقادیر موجود در حافظه را به شما نشان می‌دهد، فعال کنید.



(۴) کد را تا انتها اجرا کرده و با نحوه پر شدن خانه‌های حافظه آشنا شوید.



قسمت ۴: نتایج به دست آمده را در کادر زیر بنویسید.

اگر ورودی را به ترتیب 10 و 100 و 1000 و 10000 و 100000 قرار دهیم، در حافظه مربوط به آرایه در 20 بیت ابتدایی آن یعنی 4 بیت مختص به هر عدد در آرایه ورودی ها به شکل هگزا نوشته شده اند به طور مثال برای این 5 عدد ذخیره شده در آرایه، مموری به شکل زیر خواهد بود:

0a 00 00 00 64 00 00 00 e8 03 00 00 10 27 00 00 a0 86 01 00

برای خواندن صحیح مقدار باید ابتدا 4 بیت 4 بیت جدا کرد. سپس دو رقم سمت راست یعنی دو رقم چهارم از هر دسته چهارتایی را نوشته و دو رقم قبلی را جلوی آن و سپس دو رقم دوم و اول. یعنی برای خواندن درست مقدار پنجم آرایه باید این مقادیر را جدا کرد a0 86 01 00 سپس آنرا به شکل 000186a0 نوشت. که مقدار هگزادسیمال عدد 100000 میباشد.

## ۵. 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 نداشته، 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؟

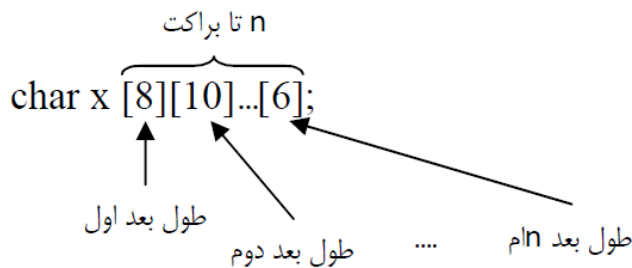
**قسمت ۵:** موارد خواسته شده را انجام دهید. نتایج به دست آمده و یافته‌های خود را در کادر زیر بنویسید.

```
//average caculation
int arr[10] = { 2,5,3,4,1,3,6,9,7,4 };
float a = 0, b = 0;
for (int i = 0; i < 10; i++){
    b = arr[i];
    a += b;
}
printf("%f", a / 10);

//کد قسمت سوم
int arr[10] = { 2,5,3,4,1,3,6,9,7,4 };
int rar[10];
for (int i = 0; i < 10; i++){
    for (int j = i; j >= 0; j--){
        if (arr[j] > arr[i]){
            rar[i] = j;
            break;
        }
        else
            rar[i] = -1;
    }
}
for (int i = 0; i < 10; i++){
    printf("%d\t", rar[i]);
}
```

## آرایه های چند بعدی:

در زبان C می توان آرایه هایی با بیش از یک بعد نیز تعریف و استفاده کرد. نحوه ی تعریف یک آرایه از نوع کاراکتر با n بعد به صورت زیر است:



**دقت کنید:** حافظه ی کامپیوتر مانند یک آرایه ی یک بعدی است. لذا برای شبیه سازی آرایه هایی با ابعاد بیش تر سطریهای 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.

← ۶. انجام دهید!

مانند قسمت ۴، برنامه را در debug mode اجرا کرده و عناصر حافظه را مشاهده 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. ضمن اینکه تفاوت هدف این قسمت با قسمت ۴ در این است که قرار است که شما در قسمت ۶ با نحوه ی ذخیره شدن داده در آرایه های دو بعدی آشنا شوید.

```
#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 ۶: نتایج به دست آمده را در کادر زیر بنویسید.

```
#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];
        }
    }

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

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

    return 0;
}
```

نوع ذخیره داده در آرایه دوبعدی همانند آرایه یک بعدی است با این تفاوت که هر سطر از آرایه دوبعدی را میتوان یک آرایه یک بعدی دانست. پس مقادیر هر سطر مانند چند آرایه یک بعدی قرار میگیرند.

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