

یادآوری: تمامی تمرینات و اطلاعات مربوط به تحویل آنها در سایت درس قرار داده میشوند:

<http://ele.aut.ac.ir/~btaheri/cpp/>

آرایه ها:

- مثال: می توان یک آرایه ی چند بعدی (در اینجا دو بعدی) را بصورت یکجا و به صورت زیر مقدار دهی کرد:

```
const int NROWS = 3;  
const int NCOLUMNS = 2;  
int a[NROWS][NCOLUMNS] = {{1}, {3}, {5}};
```

سوال برنامه ی فوق را کامل کنید و نشان دهید محتوای آرایه ی فوق بصورت زیر است:

1	0
3	0
5	0

حواس جمع!

وقتی یک آرایه را تعریف میکنیم، سائز ابعاد آن باید با یک مقدار ثابت ۱ توصیف شود یا اینکه مقادیر عددی برای توصیف ابعاد آن وارد شوند. در غیر این صورت کامپایلر ایراد میگیرد. این یک خطای ساختاری متداول است!

- یادآوری: عملگر & آدرس یک متغیر را در حافظه می دهد. لذا می توان از آن برای چاپ آدرس های آرایه استفاده کرد.

```
1. //File:ArrayDeclaration.cpp  
2. #include <iostream>  
3. using namespace std;  
4.  
5. int main(void)  
6. {  
7.     const int NROWS = 3;  
8.     const int NCOLUMNS = 2;  
9.     int a[NROWS][NCOLUMNS] = {{1}, {3}, {5}};  
10.    cout << a[0][0] << ":" << a[0][1] << "\n";  
11.    cout << a[1][0] << ":" << a[1][1] << "\n";  
12.    cout << a[2][0] << ":" << a[2][1] << "\n";  
13.    cout << "&a[2][1]: " << &a[2][1] << "\n";  
14.    cout << "&a[2][0]: " << &a[2][0] << "\n";  
15.    cout << "&a[1][1]: " << &a[1][1] << "\n";  
16.    cout << "&a[1][0]: " << &a[1][0] << "\n";  
17.    cout << "&a[0][1]: " << &a[0][1] << "\n";  
18.    cout << "&a[0][0]: " << &a[0][0] << "\n";  
19.    cout << "a: " << a << "\n";  
20.    return 0;  
21. }
```

سوال:

- با نوشتن یک تابع، و ارجاع آرایه به آن، و مقایسه ی آدرس خانه های آن در داخل و خارج تابع، نشان دهید که ارجاع آرایه به تابع به صورت Call By Reference انجام می گیرد.

کار بیشتر با رشته ها :

کلاس رشتهها را میتوان با اضافه کردن دستور `<string>` استفاده کرد.

شما تا اینجا از چنین دستوراتی استفاده کرده اید:

```
cout<< "This is a string!\n";
```

^۱ مقادیری که با `const` تعریف شده‌اند یا هر نوع ثابت دیگر.

شما در این دستور نیز بطور ناخود آگاه از رشته‌های کاراکتری برای نمایش و چاپ مطلبی استفاده کرده اید. حال می‌خواهیم با کار با این نوع داده‌ها بیشتر آشنا شویم. برای معرفی و مقدار دهی به یک متغیر رشته ای، چنین عمل می‌کنیم:

```
string str = "This is a string!\n";
```

یک رشته، آرایه ای از کاراکترهاست که به دنبال هم قرار گرفته‌اند. `str[0]` همان "T" و `str[1]` همان "h" است و....

- سوال: اگر از شما بخواهند یک رشته از ورودی بگیرید چه می‌کنید؟

به ورودی برنامه ی زیر یک جمله (مجموعه ای از کلمات با فاصله) دهید و خروجی را مشاهده کنید:

```
1. //File:string1.cpp
2. #include <iostream>
3. #include <string>
4. using namespace std;
5. int main(void)
6. {
7.     string response;
8.     cin>> response;
9.     cout<< response;
10.    return 0;
11. }
```

چه نتیجه ای در مورد عملکرد `cin` در مورد رشته های با فاصله می‌گیرید؟

حال برنامه ی زیر را اجرا کنید و به عملکرد آن توجه کنید.

```
12. //File:string1.cpp
13. #include <iostream>
14. #include <string>
15. using namespace std;
16. int main(void)
17. {
18.     string response;
19.     do
20.     {
21.         cin>> response;
22.         cout<< response + "\n";
23.     }while ('.' != response[0]); //Note vector notation
24.    return 0;
25. }
```

- با استفاده از دستور `getline()` می‌توان یک جمله شامل فاصله را از ورودی دریافت کرد:

```
1. //File:string2.cpp
2. #include <iostream>
3. #include <string>
4. using namespace std;
5. int main(void)
6. {
7.     string sentence;
8.     do
9.     {
10.         getline(cin,sentence);
11.         cout << sentence + "\n";
12.     }while ('.' != sentence[0]);
13.    return 0;
14. }
```

- تابع `length()` طول یک داده‌ی رشته‌ای را بر می‌گرداند!
- تابع `substr(m,n)` یک زیر مجموعه از رشته را بر می‌گرداند!
- می‌توان توسط عملگر `+` دو رشته را با هم جمع کرد!

```
1. //File: string3.cpp
2. #include <iostream>
3. #include <string>
4. using namespace std;
5. int main(void)
6. {
7.     cout << "Type in a string: ";
```

```

8.     string r1;
9.     cin >> r1;
10.    cout << "Your response was " << r1.length() << "chars long\n";
11.    cout << "Type in another string: ";
12.    string r2;
13.    cin >> r2;
14.    cout << "Your response was " << r2.length() << "
15.    chars long\n";
16.    cout << "Putting them together: " << r1 + " " + r2 << "\n";
17.    cout << "Slicing and splicing them up:"<< r1.
18.    substr(0,2) + r2.substr(4,7) << "\n";
19.    return 0;
20.}

```

سوال: برنامه ی فوق را دوباره خودتان بنویسید و آن را توضیح دهید!

- می توان یک رشته را در یک رشته ی دیگر کپی کرد! باور نمی کنید؟!

```

//string4
string string1 = "cpp";
string string2 = string1;
cout << string2;

```

سوال:

۱. برنامه ی فوق را تکمیل کنید!

۲. آیا چنین کاری با آرایه ها درست است؟ یعنی مثلا

```

int a[5] = {1,2,3,4,5};
int b[2] = {1,2};
b=a;

```

یا:

```

int a[5] = {1,2,3,4,5};
int b[2] = {1,2};
a=b;

```

یا:

```

int a[2] = {3,4};
int b[2] = {1,2};
a=b;

```

- تابع assign(stringName,m,n) نیز مقدار یک رشته را در رشته ی دیگر کپی می کند!

```

//File:string5
string string1 = "cpp is very strong";
string string2 = string1.assign(string1,0,3);
cout << string2;

```

سوال:

۱. برنامه ی فوق را کامل کنید و نتیجه را مشاهده کنید!

۲. در برنامه ی فوق عدد 3 نشان دهنده ی اندیس انتها است یا طول رشته ؟!

- میتوان یکی از حروف یک رشته را با حرف دیگری جایگزین کرد!!

```

1. //File:string6.cpp
2. #include <iostream>
3. #include <string>
4. using namespace std;
5. int main()
6. {
7.     string string1="HAT";
8.     string1[0]='C';
9.     cout<<string1;
10.    return 0;
11. }

```

- می توانستیم عمل فوق را با تابع at() نیز انجام دهیم!

```

//File:string7.cpp

```

```
string string1="HAT";
string1.at(0)='C';
cout<<string1;
```

سوال:

۱. برنامه ی فوق رو تکمیل کنید.

• می توان به جای + از تابع `append()` استفاده کرد!!

```
//File:string8.cpp
string string1 = "BUS";
string string2 = "STOP";
string1.append(string2);
cout << string1;
```

سوال:

۱. برنامه ی فوق را تکمیل کنید!

• برخی توابع دیگر مربوط به رشته ها می باشند که کمتر کاربرد دارند. آنها را در صفحات ۲۱۷ تا ۲۲۱ کتاب مطالعه کنید!

کمی گرافیک!!

راهنمای نصب WinBGM روی CodeBlocks قبل از روی سایت گذاشته شده! لطفا اون رو بخونید و روی کامپایلر خود اجرا کنید!

• رسم یک دایره !

```
1. //File:circle.cpp
2. #include "winbgim.h"
3. #include "winbgim.cpp"
4. int main()
5. {
6.     int i,x,y,xx;
7.     initwindow(800,600);
8.     setcolor(WHITE);
9.     circle(getmaxx()/2,getmaxy()/2,50);
10.    getch();
11.    closegraph();
12.    return 0;
13. }
```

سوال:

۱. خط ۲ و ۳ برنامه چه کاری می کنند؟ آیا می توانیم خط ۳ را حذف کنیم؟! چرا؟! خط ۲ چطور؟!

۲. خط ۷ برنامه چه کاری می کند؟

۳. خط ۸ برنامه چه کاری می کند؟

۴. خط ۹ برنامه چه کاری می کند؟

۵. خط ۱۰ و ۱۱ برنامه چه کاری می کنند؟

۶. در زیر رنگ های مختلف قابل استفاده دیده می شود. سعی کنید از چند نمونه از آنها استفاده کنید:

رنگ	نام رنگ	کد رنگ
مشکی	BLACK	0
آبی	BLUE	1
سبز	GREEN	2
آبی تیره	CYAN	3
قرمز	RED	4
صورتی	MAGENTA	5
قهوه ای	BROWN	6

7	LIGHTGRAY	خاکستری روشن
8	DARKGRAY	خاکستری تیره
9	LIGHTBLUE	آبی روشن
10	LIGHTGREEN	سبز روشن
12	LIGHTRED	قرمز روشن
13	LIGHTMAGENTA	ارغوانی روشن
14	YELLOW	زرد
15	WHITE	سفید

شما امکانات بیشتری هم در اختیار دارید. میتوانید به کمک تابع $COLOR(R,G,B)$ رنگ دلخواهی را تولید کرده و سپس رنگ تولید شده توسط را به عنوان ورودی به تابع $setcolor()$ بدهید. احتمالاً میدانید که هر رنگی را میتوان با ترکیبی از سه رنگ قرمز، سبز و آبی به وجود آورد. تابع $COLOR(R,G,B)$ به شما این امکان را میدهد تا بتوانید هر رنگی را که لازم دارید به وجود آورید. به عبارت دیگر این تابع رنگهای قرمز، سبز و آبی را با نسبتی که شما مشخص میکنید، با هم مخلوط میکند تا رنگ مورد نظر تولید شود. این تابع دارای سه آرگومان است که به ترتیب نشاندهنده نسبت رنگهای قرمز و سبز و آبی هستند. مقدار ورودی این آرگومانها میتواند عددی بین ۰ تا ۲۵۵ باشد.

به عنوان نمونه:

```
setcolor(COLOR(255,0,0));
```

سوال:

۱. آیا می توانید یک برنامه بنویسید که طیف رنگ ها را به صورتی پیوسته در خروجی نمایش دهد؟ (مثل رنگین کمان!)

• رسم یک قطاع:

```
1. //File:arc.cpp
2. /*
3. void arc(int x, int y,int startAngle,int endAngle,int radius);
4.
5. Description
6. arc draws a line in the current color, using the current
7. linestyle and thickness with radius specified by
8. radius,centrespecified by (x,y),starts from startAngle &
9. ends in endAngle
10. */
11.
12. #include "winbgim.h"
13. #include "winbgim.cpp"
14. #include <iostream.h>
15. #include <conio.h>
16.
17. int main(void)
18. {
19.
20.     int errorcode;
21.     int midx, midy;
22.     int stangle = 0, endangle = 270;
23.     int radius = 200;
24.
25.     initwindow(550,450);
26.
27.     /* read result of initialization */
28.     errorcode = graphresult();
29.     if (errorcode != grOk) /* an error occurred */
30.     {
31.
32.         cout << "Graphicerror" << grapherrormsg(errorcode) << endl;
33.
```

```

34.     cout << "Press any key to halt:";
35.     getch();
36.     exit(1);      /* terminate with an errorcode */
37. }
38.
39. midx = getmaxx() / 2;
40. midy = getmaxy() / 2;
41. setcolor(COLOR(255,255,0));
42.
43. /* draw arc */
44. arc(midx, midy, stangle, endangle, radius);
45.
46. /* clean up */
47. getch();
48. closegraph();
49. return 0;
50. }

```

سوال:

۱. نقش سطرهای ۲۸ تا ۳۷ چیست؟
۲. طرز استفاده از تابع `arc()` را توضیح دهید.

چند تابع دیگر از WinBGI به صورت زیر هستند.

۱) `clearviewport()`: این تابع `portview` را پاک و موقعیت جاری^۲ را به مختصات (0,0) منتقل میکند.

```
void clearviewport(void);
```

سوال:

۱. با نوشتن برنامه ای عملکرد این تابع را نشان دهید.

۲) `delay()`: این تابع هم از توابع مفید به شمار میرود و باعث میگردد که روند اجرای برنامه برای مدت زمان مشخصی متوقف گردد. با کمک این تابع میتوان برخی تصاویر متحرک را به وجود آورد. این تابع فقط یک ورودی دارد که آن هم مدت زمان توقف برنامه بر حسب میلیثانیه است.

```
void delay(int millisec);
```

سوال:

۱. با نوشتن برنامه ای عملکرد این تابع را نشان دهید. (مثلا یک برنامه بنویسید که در آن یک دایره از چپ به راست حرکت می کند)
۲. برنامه ی فوق را تکمیل کنید که دایره بعد از برخورد به دیواره، با همان سرعت برگردد!

۳) `drawpoly()`: از این تابع برای رسم یک چندضلعی استفاده میشود. این تابع دارای دو آرگومان ورودی است که آرگومان اول مشخص کنندهی تعداد رئوس این چند ضلعی است و آرگومان دوم نشاندهدی آرایهای است که مختصات این رئوس در آن ذخیره شدهاند.

```
void drawpoly(int numpoints, int *polypoints);
```

آرایهای که مختصات متغیرها را در خود نگهداری میکند یک آرایهی یک بعدی است. اما با توجه به اینکه مختصات هر نقطه را باید با دو عدد مشخص کرد، این مختصات به صورت متوالی در آرایه قرار میگیرند. برای مثال عضو صفر آرایه برابر است با مقدار x نقطهی اول و عضو یک آرایه برابر است با مقدار y نقطهی اول و... . برنامهی زیر طرز استفاده از این تابع را نشان میدهد.

```

//File:poly.cpp
#include "winbgim.h"
#include "winbgim.cpp"
#include <stdlib.h>

```

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main(void)
{
    int errorcode;
    int maxx, maxy;
    int poly[10]; /* our polygon array */
    /* initialize graphics and local variables */
    initwindow(800,500);
    /* read result of initialization */
    errorcode = graphresult();
    if (errorcode != grOk) { /* an error occurred */

        printf("Graphics error: %s\n", grapherrormsg(errorcode));

        printf("Press any key to halt:");

        getch();

        exit(1); /* terminate with an error code */
    }
    maxx = getmaxx();
    maxy = getmaxy();
    poly[0] = 0; /* first vertex */
    poly[1] = maxy/2;
    poly[2] = maxx - 20; /* second vertex */
    poly[3] = 20;
    poly[4] = maxx - 50; /* third vertex */
    poly[5] = maxy - 20;
    poly[6] = maxx / 2; /* fourth vertex */
    poly[7] = maxy / 2;
    poly[8] = poly[0]; /* drawpoly doesn't automatically close */
    poly[9] = poly[1]; /* the polygon, so we close it */
    drawpoly(5, poly); /* draw the polygon */
    /* clean up */
    getch();
    closegraph();
    return 0;
}

```

سوال:

۱. برنامه ی فوق را اجرا کنید و عملکرد آن را توضیح دهید.
۲. با استفاده از برنامه ی فوق یک ستاره رسم کنید.
۳. با استفاده از برنامه ی فوق، آرم اسرائیل را رسم کنید. (توجه! این مثال فقط جنبه ی آموزشی دارد!)

۴ - ellipse(): از این تابع میتوان برای رسم یک بیضی و یا کمانی از یک بیضی استفاده نمود.

```

void ellipse(int x, int y, int stangle, int endangle, int xradius,
int yradius);

```

دو آرگومان اول این تابع مختصات مرکز بیضی هستند. آرگومان سوم زاویهی شروع، آرگومان چهارم زاویه پایان، آرگومان پنجم شعاع در راستای محور X و آرگومان ششم شعاع در راستای محور Y است.

سوال:

۱. با نوشتن برنامه ای عملکرد تابع فوق را نشان دهید.

۵ - `fillellipse()`: از این تابع میتوان برای رسم یک بیضی توپر استفاده نمود. رنگ داخل بیضی به رنگ قلمجاری درخواهد آمد. دو آرگومان اول مرکز بیضی، آرگومان سوم شعاع آن در راستای محور x و آرگومان آخر شعاع در راستای محور y است.

```
void fillellipse(int x, int y, int xradius, int yradius);
```

سوال:

۱. با نوشتن برنامه ای عملکرد تابع فوق را نشان دهید.

۶ - `fillpoly()`: این تابع هم مانند `drawpoly` است، با این تفاوت که داخل چندضلعی رسم شده را رنگ خواهد کرد.

```
void fillpoly(int numpoints, int *polypoints);
```

سوال:

۱. با نوشتن برنامه ای عملکرد تابع فوق را نشان دهید.

۷ - `floodfill()`: این تابع برای رنگکردن داخل یک محدوده بسته به کار میرود. برای مثال اگر قبلاً یک دایره رسم شده باشد و یک نقطه از داخل این دایره به عنوان ورودی به تابع داده شود این تابع کل محدوده داخل دایره را رنگ خواهد نمود. آرگومان ورودی سوم این تابع رنگ مرزهای قسمتی است، که باید رنگ گردد. به عبارت دیگر این تابع تمام محدوده پیرامون نقطه‌ی ورودی را رنگ میکند تا به مرز برسد. رنگ مرز توسط آرگومان ورودی سوم تابع مشخص میگردد.

```
void floodfill(int x, int y, int border);
```

```
1. //File:floodfill.cpp
2. #include "winbgim.h"
3. #include "winbgim.cpp"
4. #include <conio.h>
5. #include<iostream.h>
6. int main(void)
7. {
8.     int errorcode;
9.     int maxx, maxy;
10.    /* initialize graphics and local variables */
11.
12.    initwindow(800,600);
13.
14.    /* read result of initialization */
15.
16.    errorcode = graphresult();
17.    if (errorcode != grOk) { /* an error occurred */
18.        cout<<"Graphics error"<<grapherrormsg(errorcode);
19.        cout<<"Press any key to halt:";
20.        getch();
21.        exit(1); /* terminate with an error code */
22.    }
23.
24.    maxx = getmaxx();
25.    maxy = getmaxy();
26.
27.    /* select drawing color */
28.
29.    setcolor(WHITE);
30.
31.    /* select fill color */
32.
33.    setfillstyle(SOLID_FILL, YELLOW);
34.
35.    /* draw a border around the screen */
36.
37.    rectangle(0, 0, maxx, maxy);
38.
39.
40.    /* draw some circles */
41.
42.    circle(maxx / 3, maxy / 2, 50);
```



```

43. circle(maxx / 2, 20, 100);
44. circle(maxx-20, maxy-50, 75);
45. circle(20, maxy-20, 25);
46.
47. /* wait for a key */
48.
49. getch();
50.
51. /* fill in bounded region */
52.
53. floodfill(2, 2, WHITE);
54.
55. /* clean up */
56.
57. getch();
58. closegraph();
59. return 0;
60. }

```

سوال:

۱. عملکرد برنامه ی فوق را توضیح دهید!

۸- `linere1()`: این تابع خطی را از نقطه‌ی جاری، به نقطه‌های که، به اندازه‌ی مشخصی با نقطه‌ی جاری فاصله دارد رسم میکند.

```
void linerel(int dx, int dy);
```

- در واقع، این تابع خطی را از نقطه‌ی جاری، به نقطه‌ی مقصد میکشد. نقطه‌ی مقصد نقطه‌های است که تفاضل x آن با نقطه جاری برابر با dx و تفاوت y آن با نقطه جاری برابر با dy است.

سوال:

۱. با نوشتن برنامه ای عملکرد تابع فوق را نشان دهید.

۹- `lineto()`: با کمک این تابع میتوان خطی از نقطه‌ی جاری به نقطه‌ی x, y رسم نمود.

```
void lineto(int x, int y);
```

سوال:

۱. با نوشتن برنامه ای عملکرد تابع فوق را نشان دهید.

۱۰- `putpixel()`: این تابع پیکسلی را به مختصات x, y و با رنگ داده شده روشن میکند.

```
void putpixel(int x, int y, int color);
```

سوال:

۱. با نوشتن برنامه ای عملکرد تابع فوق را نشان دهید.

۱۱- `rectangle()`: از این تابع میتوان برای رسم مربع یا مستطیل استفاده نمود!

```
void rectangle(int left, int top, int right, int bottom);
```

سوال:

۱. با نوشتن برنامه ای عملکرد تابع فوق را نشان دهید.

- تعدادی توابع دیگر در این کتابخانه وجود دارند که برای آشنایی با آنها به صفحات ۶۲۰ تا ۶۲۶ مراجعه کنید.