اصول کامپیوتر ۱

مبانی کامپیوتر و برنامه‌سازی

«جلسه‌ی هشتم»

دانشکده‌ی علوم ریاضی - دانشگاه شهید بهشتی

نیم‌سال اول ۹۰-۱۳۸۹

مدرس: سید علی کتان‌فروش

گردآوری و تألیف: سید علی کتان‌فروش

حق هر نوع استفاده متعلق به پدیدآورنده است.

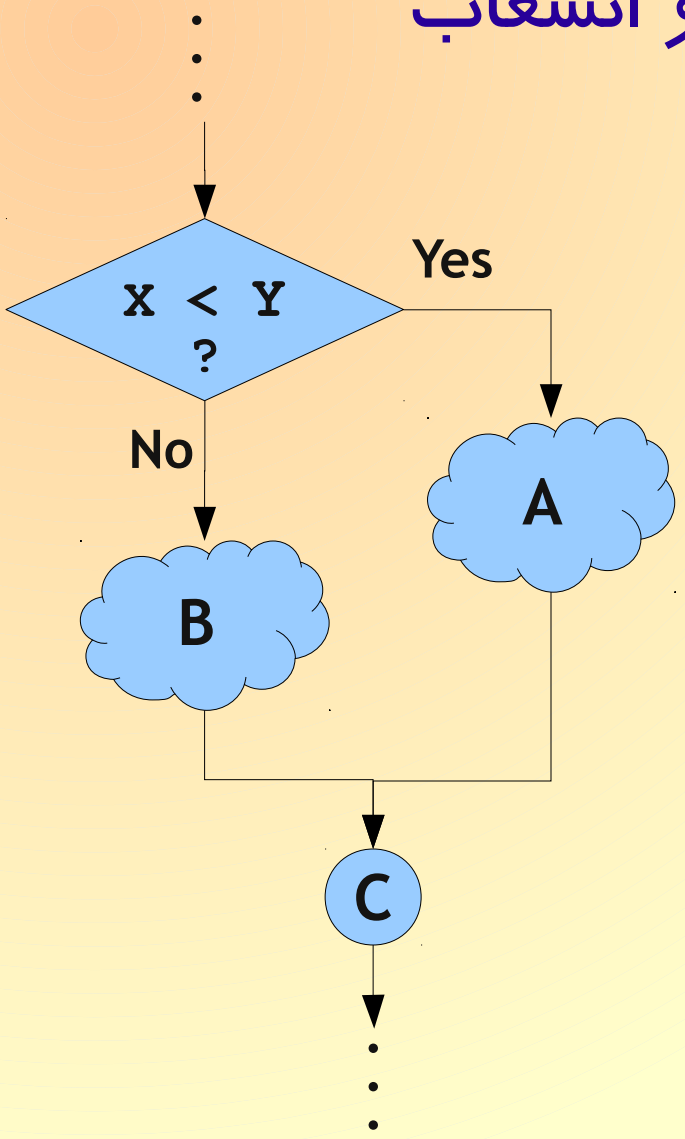
تصمیم‌گیری و انشعاب

Decision and Branch

یادآوری

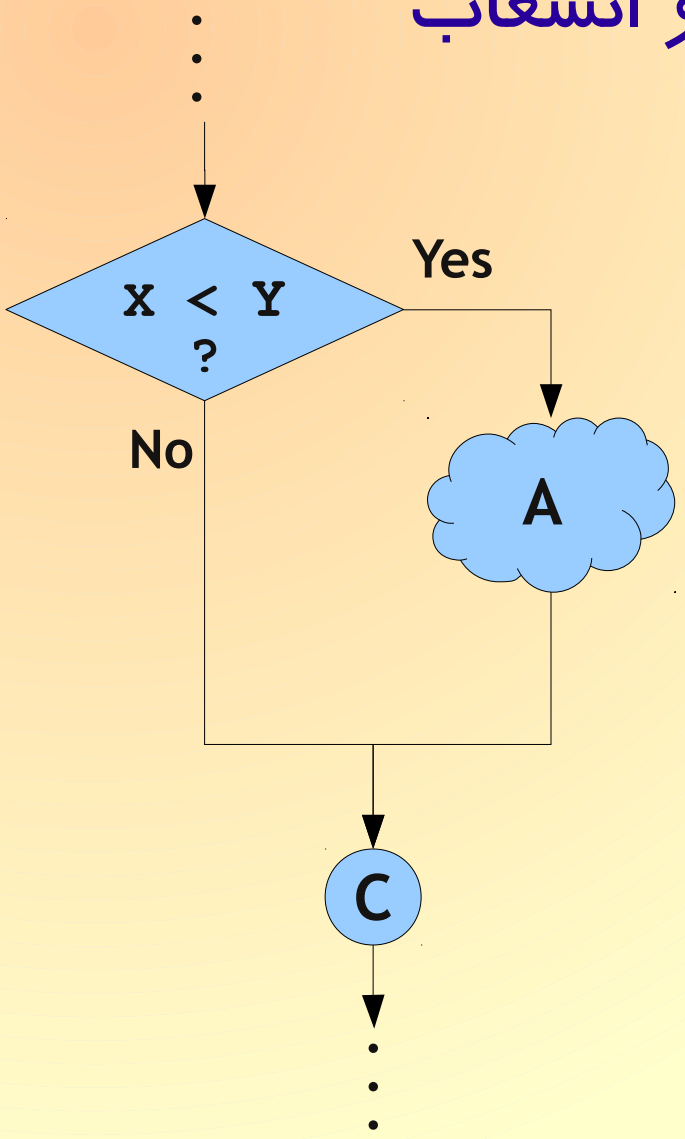
- در جلسه‌ی قبل با الگوی «تصمیم‌گیری و انشعاب» آشنا شدیم.
- در الگوی «تصمیم‌گیری و انشعاب»، اجرای الگوریتم بسته به آنکه عبارتی شرطی، ارزش درست یا نادرست به خود بگیرد به دو مسیر منشعب می‌شود.

تصمیم‌گیری و انشعاب




```
if ( X < Y ) {
 A
}
else {
 B
}
C
```

تصمیم گیری و انشعاب



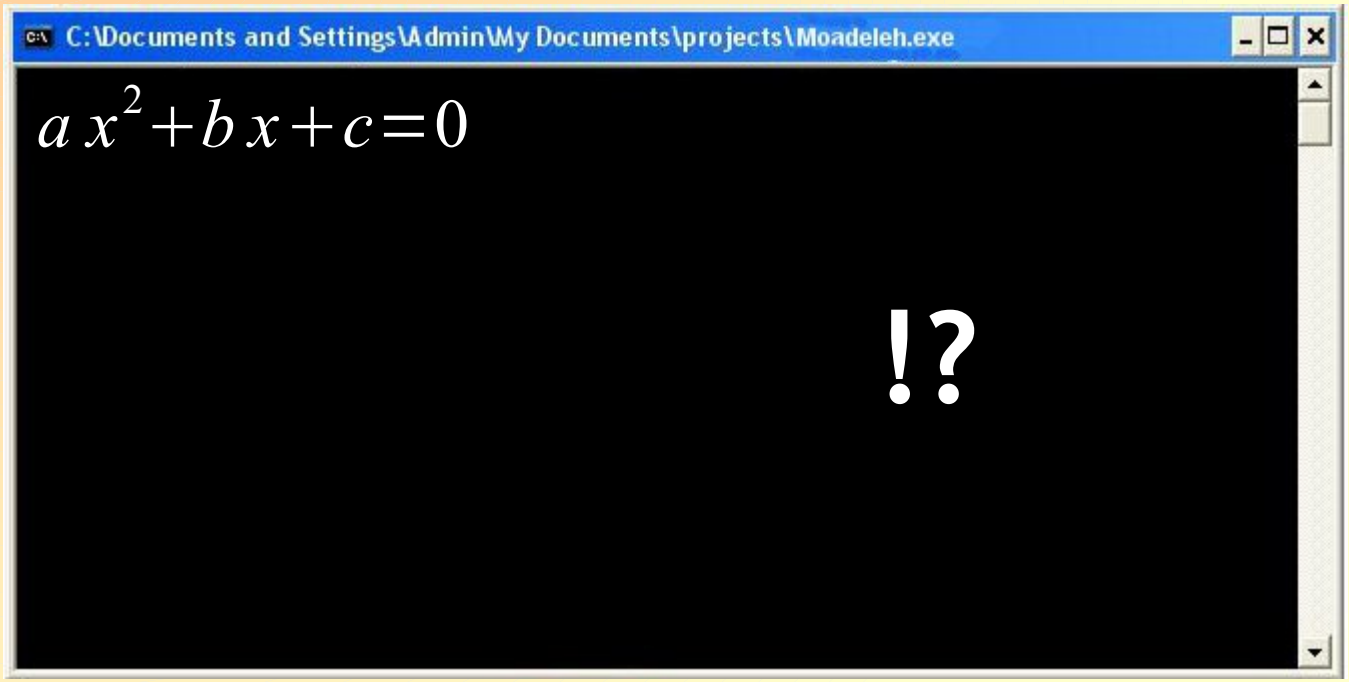

```
if ( X < Y ) {
 A
}
C
```

مثال.

الگوریتمی برای حل و بحث معادله‌ی درجه دو ارائه دهید.

$$a\,x^2+b\,x+c=0$$

- **کاربر معادله را چگونه وارد می‌کند ؟**
- **به چه شکلی جواب و بحث آن نمایش داده می‌شوند ؟**


C:\Documents and Settings\Admin\My Documents\projects\Moadeleh.exe
$a x^2 + b x + c = 0$
!?

/scratch/pfrauenf/concepts-cvs/hp/mathematica/hexaeder.nb

File Edit Cell Format Input

/u/u2/pfrauenf/mathematica/AnalysisII-Sperb-SS01/U9.nb

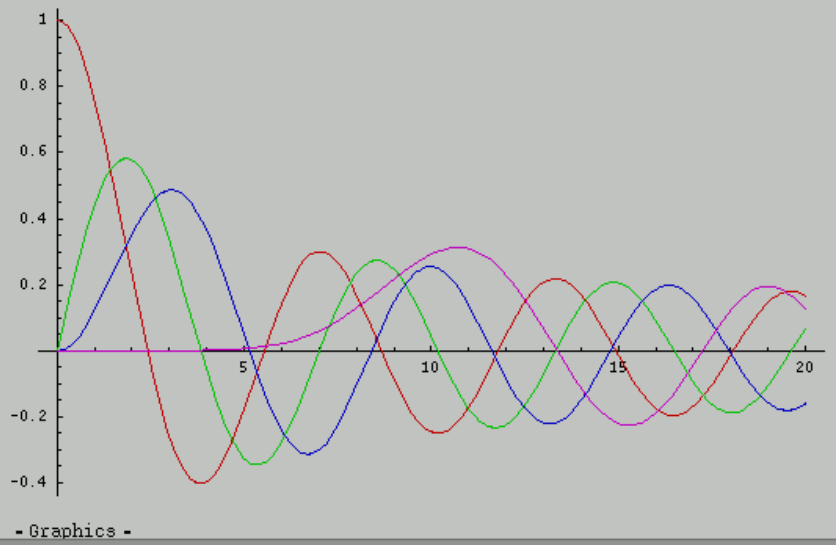
File Edit Cell Format Input Kernel Find Window Help

BesselJ[n, z] gives the Bessel function of the first kind $J_n(z)$.

$J_n(z)$ satisfies the differential equation $z^2 y'' + z y' + (z^2 - n^2) y = 0$.

BesselJ[n, z] has a branch cut discontinuity in the complexzplane running from $-\infty$ to 0.

```
Plot[{BesselJ[0, x], BesselJ[1, x], BesselJ[2, x], BesselJ[9, x]},
 {x, 0, 20},
 PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 1, 0], RGBColor[0, 0, 1],
   RGBColor[1, 0, 1]}]
```



- Graphics -

Help Browser

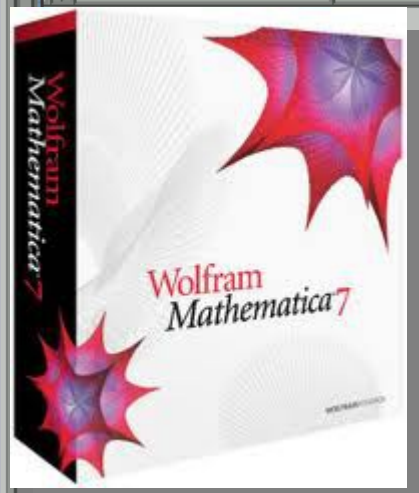
Help

thematica Book

Index

<< Go Back Hide Categories

Plot3D
ListPlot3D
ParametricPlot3D




```
aa = Table[A[[i, j]],
bb = Table[B[[i, j]], {
ll = Table[L[[i]], {i,
mm = Table[M[[i]], {i,
ss = LinearSolve[aa, l
(ll + mm)[[3]] / (aa + bb
```

$$\begin{pmatrix} 0. \\ 0. \\ 0. \\ 0. \\ 0. \end{pmatrix}$$

$$\begin{pmatrix} 0. \\ 0. \\ 1. \\ 0. \end{pmatrix}$$

0.4

p=2: Neumann-Randintegrale

Auf jedem Face kommen die
grade. Daher: nur x = 1 wird be

```
nE[1][y_, z_] = y * (1 -
nE[2][y_, z_] = z * (1 -
nE[3][y_, z_] = z * (1 -
nE[4][y_, z_] = y * (1 -
For[i = 1, i ≤ 4, i++,
 Print[
  Integrate[Integrate
    {y, 0, 1}], {z, 0.
]
```

0.0138889

0.0138889

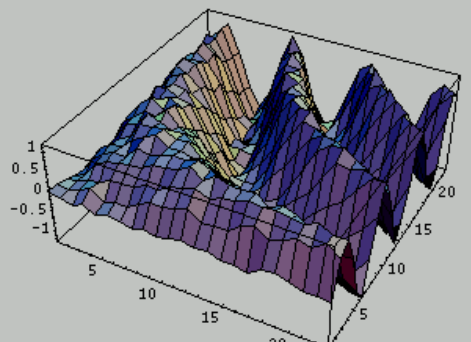
0.0277778

0.0277778

```
nF[y_, z_] = y * (1 - y) * z * (1 - z);
Integrate[Integrate[nF[y, z] * f[1, y, z], {y, 0, 1}],
  {z, 0, 1}] // N
```

0.00694444

In[1]:=
```
ListPlot3D[Table[Sin[x y] + Random[Real, {-0.15, 0.15}],
  {x, 0, 3 Pi/2, Pi/15}, {y, 0, 3 Pi/2, Pi/15}]];
```



Wolfram Mathematica 7.0 - [Untitled-1 *]

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

Untitled-1 *

In[6]:= Solve [a x^2 + b x + c == 0 , x]

Out[6]= $\left\{\left\{x \to \frac{-b - \sqrt{b^2 - 4\,a\,c}}{2\,a}\right\}, \left\{x \to \frac{-b + \sqrt{b^2 - 4\,a\,c}}{2\,a}\right\}\right\}$

In[7]:= Solve [3 x^2 - 4 x + 1 == 0]

Out[7]= $\left\{\left\{x \to \frac{1}{3}\right\}, \{x \to 1\}\right\}$

حل نمادی

Symbolic solution

In[8]:= Solve [1 x^2 + 6 x + 9 == 0 , x]

Out[8]= {{x → -3}, {x → -3}}

حل عددی

Numeric solution

In[10]:= Solve [2 x^2 + 4 x + 1 == 0 , x]

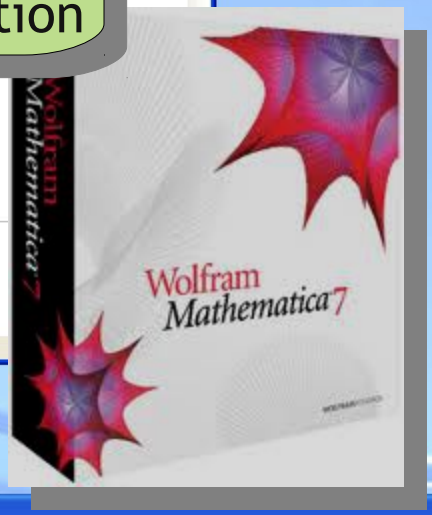
Out[10]= $\left\{\left\{x \to \frac{1}{2}\left(-2 - \sqrt{2}\right)\right\}, \left\{x \to \frac{1}{2}\left(-2 + \sqrt{2}\right)\right\}\right\}$

ریشه‌ی موهومی

Imaginary solution

In[11]:= Solve [x^2 + 2 x + 3 == 0 , x]

Out[11]= $\{\{x \to -1 - i\sqrt{2}\}, \{x \to -1 + i\sqrt{2}\}\}$



start

Wolfram Mathematica...

Adobe Photoshop CS...

مثال. الگوریتمی برای حل و بحث معادله‌ی درجه دو ارائه دهید.

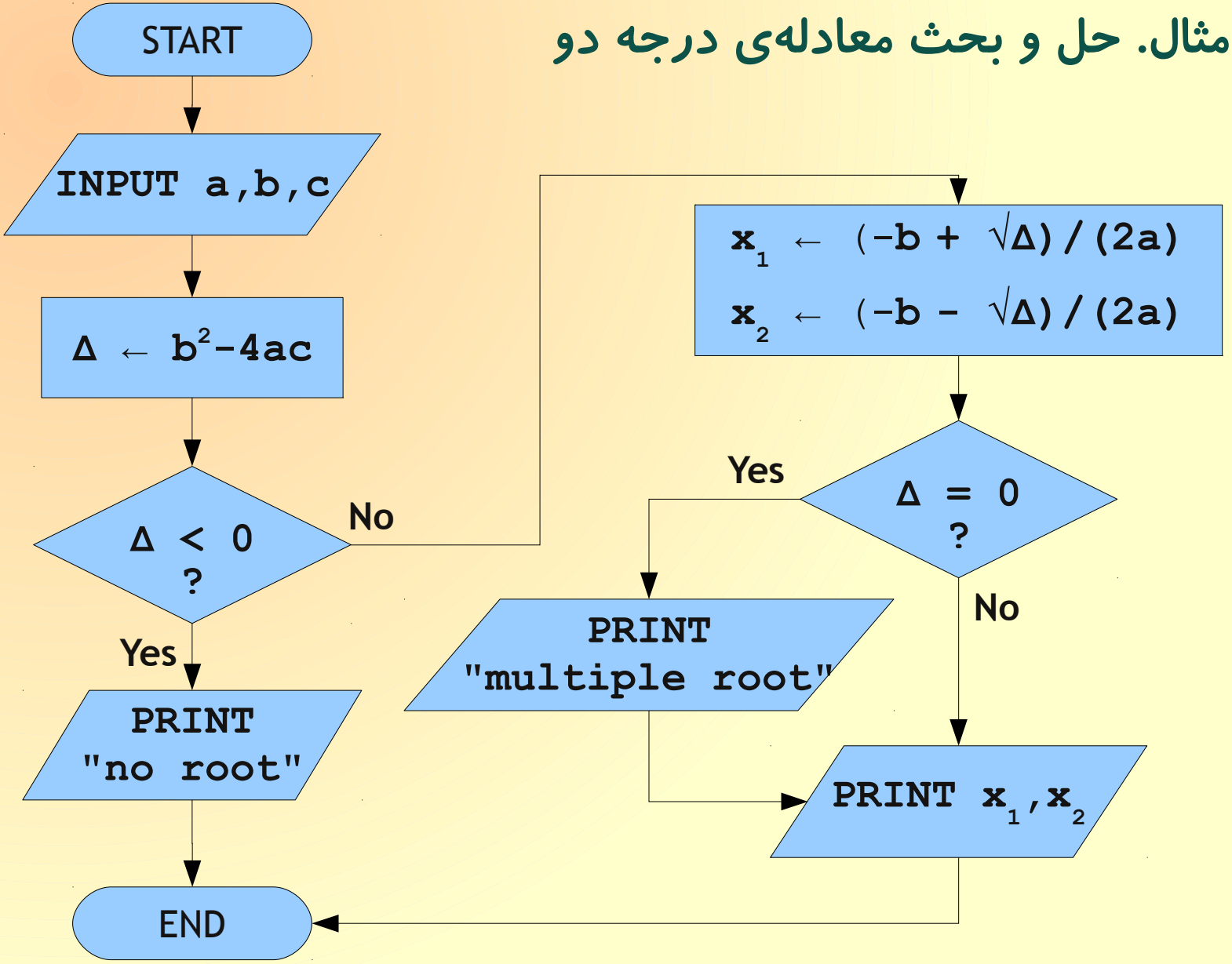
- نرم‌افزارهای قدرتمندی مثل *Mathematica* این امکان را به کاربر می‌دهند که معادله را با ظاهری نزدیک به نگارش دستنویس وارد کنند و جواب را برحسب ضرایب به صورت پارامتری دریافت کنند.
- با این حال برای حل عددی یک معادله‌ی درجه دو، تنها چیزی که نیاز است مقدار ضرایب معادله است.
- با این استدلال لازم است مسأله را به صورت زیر درنظر گیریم ...

مثال. الگوریتمی ارائه دهید که ضرایب یک معادله‌ی درجه دوم را از کاربر دریافت کند و ریشه‌های آنرا در صورت وجود چاپ کند و در غیراینصورت پیام مناسب نمایش دهد.

هنوز نمی‌توانیم برنامه‌ای بنویسیم که در آن این امکان برای کاربر وجود داشته باشد که ضرایب را به صورت زیر وارد کند.

باز برای سادگی بیشتر، قرارداد می‌کنیم که کاربر ضرایب را به ترتیب نزولی برحسب توان وارد می‌کند.

مثال. حل و بحث معادله‌ی درجه دو



```
int main( int argc, char *argv[]){
  double a,b,c;
  cin >> a >> b >> c;
  double delta;
  delta = b*b - 4*a*c;
  if ( delta < 0 )
    cout << "no root exists." << endl;
  else {
    double x1,x2;
    x1 = (-b+sqrt(delta))/(2*a);
    x2 = (-b-sqrt(delta))/(2*a);
    if ( delta == 0 )
      cout << "multiple roots" << endl;
    cout << "x1=" << x1 << " , x2=" << x2;
    cout << endl;
  }
  ....
```

```
int main( int argc, char *argv[]){
  double a,b,c;
  cin >> a >> b >> c;
  double delta;
  delta = b*b - 4*a*c;
  if ( delta < 0 )
    cout << "no root exists." << endl;
  else {
    double x1,x2;
    x1 = (-b+sqrt(delta))/(2*a);
    x2 = (-b-sqrt(delta))/(2*a);
    if ( delta == 0 )
      cout << "multiple roots" << endl;
    cout << "x1=" << x1 << " , x2=" << x2;
    cout << endl;
  }
  ....
```

ضرایب معادله، Δ و ریشه‌ها از نوع اعداد اعشاری با دقت مضاعف هستند.

```
int main( int argc, char *argv[]){
  double a,b,c;
  cin >> a >> b >> c;
  double delta;
  delta = b*b - 4*a*c;
  if ( delta < 0 )
    cout << "no root ex
  else {
    double x1,x2;
    x1 = (-b+sqrt(delta))/(2*a);
    x2 = (-b-sqrt(delta))/(2*a);
    if ( delta == 0 )
      cout << "multiple roots" << endl;
    cout << "x1=" << x1 << " , x2=" << x2;
    cout << endl;
  }
  ....
```

فقط استفاده از حروف الفبای انگلیسی و ارقام برای نامگذاری متغیرها مجاز است.

```
int main( int argc, char *argv[]){
  double a,b,c;
  cin >> a >> b >> c;
  double delta;
  delta = b*b - 4*a*c;
  if ( delta < 0 )
    cout << "no root exists." <<
  else {
    double x1,x2;
    x1 = (-b+sqrt(delta))/(2*a);
    x2 = (-b-sqrt(delta))/(2*a);
    if ( delta == 0 )
      cout << "multiple roots" << endl;
    cout << "x1=" << x1 << " , x2=" << x2;
    cout << endl;
  }
  ....
```

تابع جذر تعریف شده در cmath

```
int main( int argc, char *argv[]){
  double a,b,c;
  cin >> a >> b >> c;
  double delta;
  delta = b*b - 4*a*c;
  if ( delta < 0 )
    cout << "no root
  else {
    double x1,x2;
    x1 = (-b+sqrt(del
    x2 = (-b-sqrt(del
    if ( delta == 0 )
      cout << "multiple roots" << endl;
    cout << "x1=" << x1 << " , x2=" << x2;
    cout << endl;
  }
  ....
```

هر چند در کتابخانه‌ی cmath تابعی برای محاسبه‌ی توان وجود دارد اما در مواقعی که توان عدد صحیح و کوچکی است بهتر است بجای استفاده از آن، عمل ضرب را تکرار کنید.

مثلاً b*b بجای b^2

```
int main( int argc, char *argv
  double a,b,c;
  cin >> a >> b >> c;
  double delta = b*b - 4*a*c;
  if ( delta < 0 )
    cout << "no root exists." << endl;
  else {
    double x1,x2;
    x1 = (-b+sqrt(delta))/(2*a);
    x2 = (-b-sqrt(delta))/(2*a);
    if ( delta == 0 )
      cout << "multiple roots" << endl;
    cout << "x1=" << x1 << " , x2=" << x2;
    cout << endl;
  }
  ....
```

اعلان متغیر و اولین مقداردهی را می‌توان در یک دستور ترکیب کرد.

```
int main( int argc, char *argv[]){
  double a,b,c;
  cin >> a >> b >> c;
  double delta = b*b - 4*a*c;
  if ( delta < 0 )
    cout << "no root exists." << endl;
  else {
    double x1,x2;
    x1 = (-b+sqrt(delta))/2*a;
    x2 = (-b-sqrt(delta))/2*a;
    if ( delta == 0 )
      cout << "multiple roots" << endl;
    cout << "x1=" << x1 << " , x2=" << x2;
    cout << endl;
  }
  ....
```

حذف پرانتزها در اینجا نتیجه‌ی نامطلوب بدست می‌آورد.

مثال. الگوریتمی برای حل و بحث دستگاه خطی دو معادله‌ی دو مجهولی ارائه دهید.

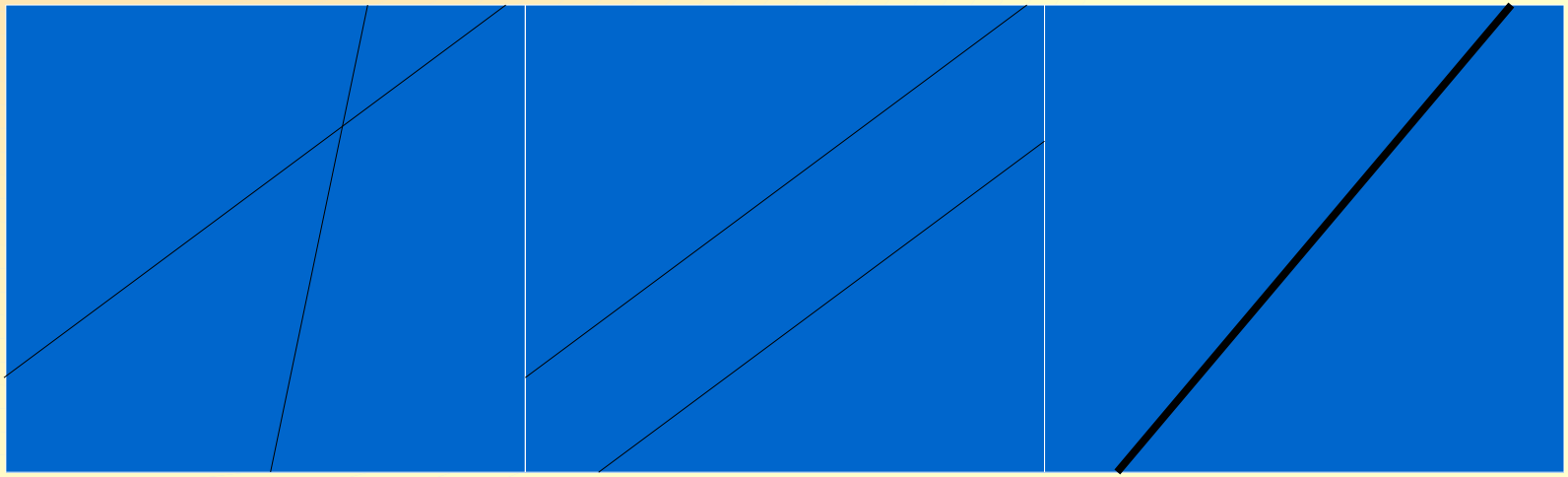
$$\begin{cases} a_1 x + b_1 y = c_1 \\ a_2 x + b_2 y = c_2 \end{cases}$$

- همانند مثال قبل، لازم است مسأله را با جزئیات دقیق‌تری تعریف کنیم.
- برای حل عددی هر دستگاه خطی دو معادله‌ی دو مجهولی، تنها داشتن ضرایب دستگاه لازم است.

مثال. الگوریتمی ارائه دهید که ضرایب یک دستگاه خطی دو معادله‌ی دو مجهولی را از کاربر دریافت کند و در صورت وجود جواب دستگاه را چاپ کند. در غیر اینصورت پیام مناسب نمایش داده شود.

$$\begin{cases} a_1 x + b_1 y = c_1 \\ a_2 x + b_2 y = c_2 \end{cases}$$

دو خط منطبق (بیشمار جواب) دو خط موازی (بدون جواب) دو خط متقاطع (یک جواب)



- شیب خط = a/b - عرض از مبداء = c/b

- دو خط منطبق هستند اگر هم شیب خط و هم عرض از مبداءشان برابر باشد.

$$\frac{a_1}{b_1}=\frac{a_2}{b_2} \quad \text{and} \quad \frac{c_1}{b_1}=\frac{c_2}{b_2}$$

- دو خط موازیند اگر شیب برابر و عرض از مبداء نابرابر داشته باشند.

$$\frac{a_1}{b_1}=\frac{a_2}{b_2} \quad \text{and} \quad \frac{c_1}{b_1}\neq\frac{c_2}{b_2}$$

- در غیر موارد فوق، دو خط یکدیگر را در یک نقطه قطع می‌کنند.

```
int main( int argc, char *argv[]){
  double a1,b1,c1;
  double a2,b2,c2;
  cout << "Enter coefficients of equation 1:";
  cin >> a1 >> b1 >> c1;
  cout << "Enter coefficients of equation 2:";
  cin >> a2 >> b2 >> c2;
  if ( a1/b1 == a2/b2 && c1/b1 == c2/b2 )
    cout << "Identical lines." << endl;
  if ( a1/b1 == a2/b2 && c1/b1 != c2/b2)
    cout << "Parallel lines." << endl;
  if ( a1/b1 != a2/b2 && c1/b1 != c2/b2) {
    double x , y;
    x = ....;
    y = ....;
    ....
  }
```

```
int main( int argc, char *argv[]){
  double a1,b1,c1;
  double a2,b2,c2;
  cout << "Enter coefficients of equation 1:";
  cin >> a1 >> b1 >> c1;
  cout << "Enter coefficients of equation 2:";
  cin >> a2 >> b2 >> c2;
  if ( a1/b1 == a2/b2 && c1/b1 == c2/b2 )
    cout << "Identical lines." << endl;
  if ( a1/b1 == a2/b2 && c1/b1 != c2/b2)
    cout << "Parallel lines." << endl;
  if ( a1/b1 != a2/b2 && c1/b1 != c2/b2) {
    double x , y;
    x = ....;
    y = ....;
    ....
  }
```

از نظر منطقی بدیهی است که تنها یکی از این سه شرط ارضاء می‌شود.

```
int main( int argc, char *argv[]){
  double a1,b1,c1;
  double a2,b2,c2;
  cout << "Enter coefficie
  cin >> a1 >> b1 >> c1;
  cout << "E          coefficients          uation 2:";
  cin >> a          >> c2;
  if ( a1/b1 == a2/b2 && c1/b1 == c2/b2 )
    cout << "I          ical lines." <<      l;
  else {
    if ( a1/b1 == a2/b2 && c1/b1 != c2/b2)
      cout << "Parallel lines." << endl;
    else
      if ( a1/b1 != a2/b2 && c1/b1 != c2/b2) {
        ....
      }
  }
```

اگر حداقل یکی از A یا B نادرست باشد آنگاه اگر A درست باشد پس حتماً B نادرست است.

A

B

A

~B

```
int main( int argc, char *argv[]){
  double a1,b1,c1;
  double a2,b2,c2;
  cout << "Enter coefficients of equation 1:";
  cin >> a1 >> b1 >> c1;
  cout << "Enter coefficients of equation 2:";
  cin >> a2 >> b2 >> c2;
  if ( a1/b1 == a2/b2 && c1/b1 == c2/b2 )
    cout << "Identical lines." << endl;
  else {
    if ( a1/b1 == a2/b2 )
      cout << "Parallel lines." << endl;
    else {
        double x,y;
        ....
    }
  }
```

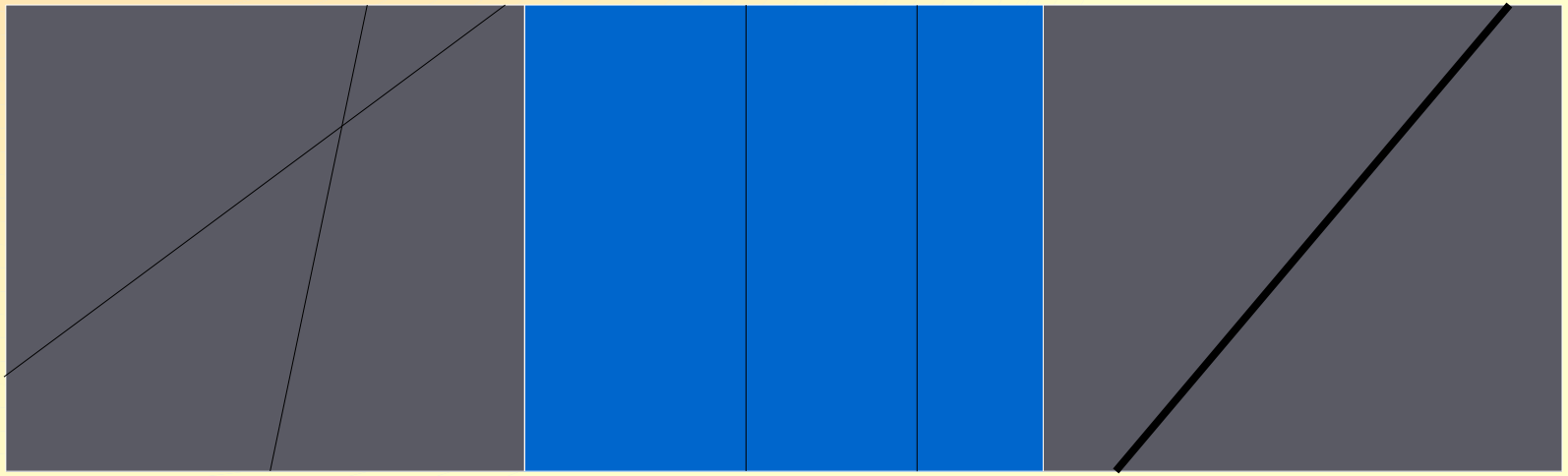
```
int main( int argc, char *argv[]){
  double a1,b1,c1;
  double a2,b2,c2;
  cout << "Enter coefficients of equation 1:";
  cin >> a1 >> b1 >> c1;
  cout << "Enter coefficients of equation 2:";
  cin >> a2 >> b2 >> c2;
  if ( a1/b1 == a2/b2 && c1/b1 == c2/b2 )
    cout << "Identical lines." << endl;
  else if ( a1/b1 == a2/b2 )
      cout << "Parallel lines." << endl;
  else {
      double x,y;
      ....
  }
  ....
```

اگر b_1 یا b_2 برابر صفر باشد برنامه با خطای تقسیم بر صفر مواجه می‌شود.

در حالیکه دستگاهی مانند مثال زیر منطقاً قابل قبول و نشاندهنده‌ی دو خط موازی است.

$$\begin{cases} x=2 \\ 2x=3 \end{cases}$$

دو خط منطبق (بیشمار جواب) دو خط موازی (بدون جواب) دو خط متقاطع (یک جواب)



روش کرامر.

- ابتدا دترمینان ماتریس ضرایب و دترمینان ماتریس‌هایی که با تعویض ضرایب معلوم با ضرایب مجهول بدست می‌آیند را محاسبه می‌کنیم:

$$\Delta = det\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}$$

$$\Delta_x = det\begin{bmatrix} c_1 & b_1 \\ c_2 & b_2 \end{bmatrix}$$

$$\Delta_y = det\begin{bmatrix} a_1 & c_1 \\ a_2 & c_2 \end{bmatrix}$$

ادامه‌ی روش کرامر.

- اگر $\Delta \neq 0$ دستگاه یک جواب دارد که به صورت زیر بدست می‌آید:

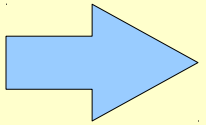
$$x = \frac{\Delta_x}{\Delta} \quad , \quad y = \frac{\Delta_y}{\Delta}$$

- در غیر اینصورت اگر $\Delta_x = 0$ آنگاه $\Delta_y = 0$ و دستگاه بیشمار جواب دارد. (دو خط منطبق بر هم).
- در غیر اینصورت دستگاه جواب ندارد. (دو خط موازی)

```
int main( int argc, char *argv[]){
  double a1,b1,c1;
  cout << "Enter coefficients of equation 1:";
  cin >> a1 >> b1 >> c1;

  double a2,b2,c2;
  cout << "Enter coefficients of equation 2:";
  cin >> a2 >> b2 >> c2;

  double delta = a1*b2-a2*b1;
  double deltaX = c1*b2-c2*b1;
  double deltaY = a1*c2-a2*c1;
```



```
  // continued from previous slide

  if ( delta != 0 ) {
    double x = deltaX/delta;
    double y = deltaY/delta;
    cout << "Lines intersect at ("
         << x << "," << y << ")" << endl;
  }
  else if ( deltaX == 0 )
     cout << "Identical lines" << endl;
  else
     cout << "Parallel lines" << endl;

  system("pause");
  return EXIT_SUCCESS;
}
```