اصول کامپیوتر ۱

مبانی کامپیوتر و برنامه‌سازی

«جلسه‌ی هفدهم»

دانشکده‌ی علوم ریاضی - دانشگاه شهید بهشتی

نیم‌سال اول ۹۰-۱۳۸۹

مدرس: سید علی کتان‌فروش

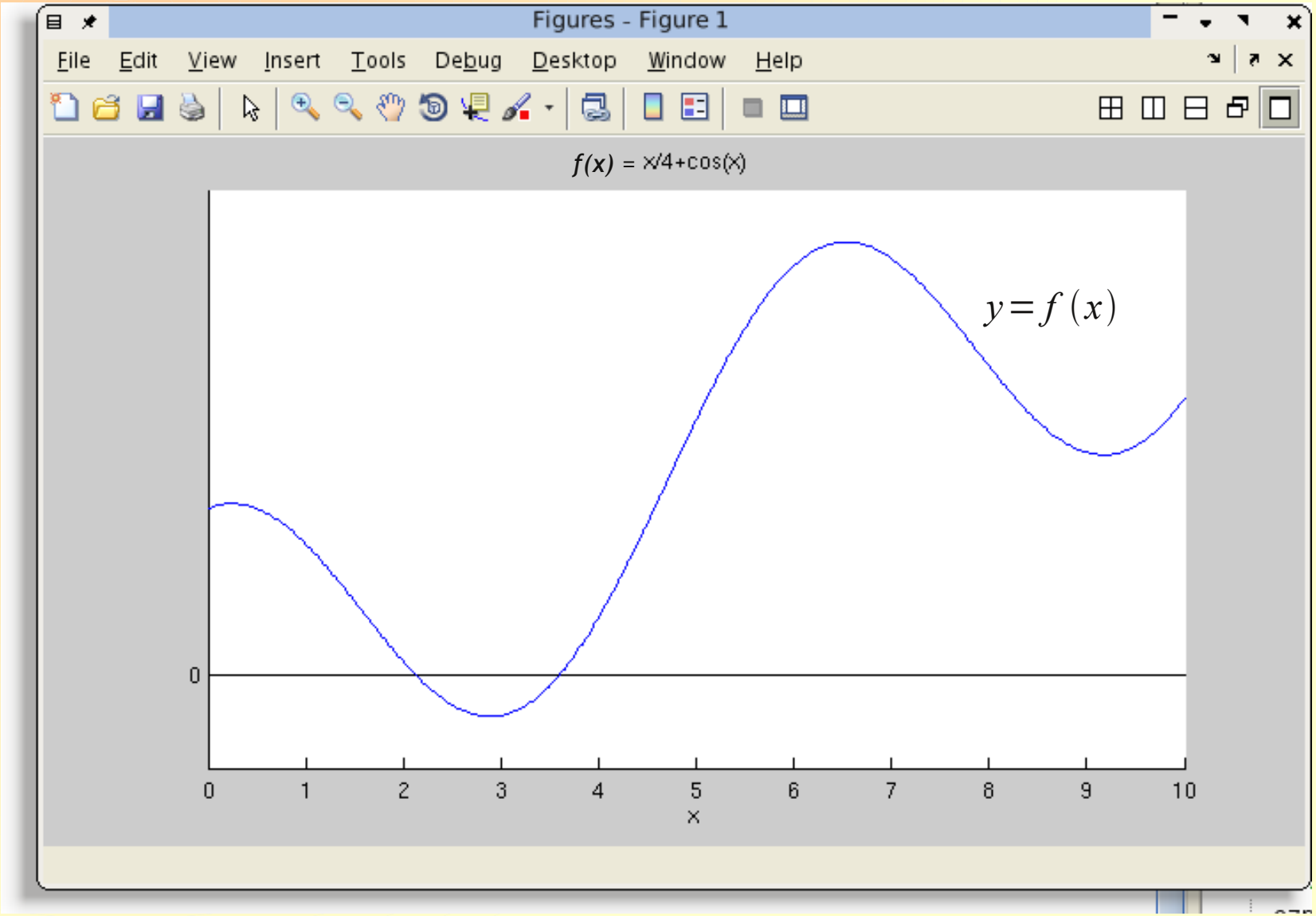
تعریف توابع در C/C++

- تابع در زبان برنامه‌نویسی C/C++، زیربرنامه‌ای است که در آن الگوریتمی مستقل از دیگر بخش‌های برنامه، پیاده‌سازی شده است.
- الگوریتم یک تابع اصولاً، به ازای یک یا چند متغیر پارامتر تعریف می‌شود.
- اساساً می‌بایست بین توابع ریاضی و توابع برنامه‌نویسی تمایز قائل شوید. هرچند معمولاً می‌توان توابع ریاضی را در قالب یک تابع برنامه‌نویسی تعریف کرد.

تعریف توابع در C/C++

برای تعریف یک تابع جدید در برنامه، نکات زیر را باید مورد توجه قرار دهید.

- شناسه‌ی تابع (Function identifier).
- پارامترها یا آرگومان‌های تابع (Function arguments).
- نوع مقدار بازگشتی تابع.


Figures - Figure 1
File Edit View Insert Tools Debug Desktop Window Help
f(x) = x/4+cos(x)
y = f(x)
0
0 1 2 3 4 5 6 7 8 9 10
x

تعریف توابع در C/C++

مثال ۱. تابع ریاضی $f(x)=x/4+\cos(x)$ را در قالب یک تابع برنامه‌نویسی پیاده‌سازی کنید.

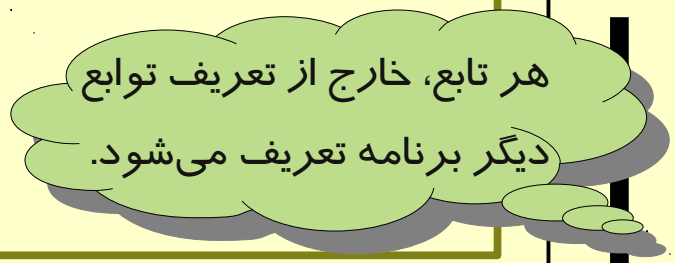
- **شناسه‌ی تابع: `f`**
- **آرگومان تابع: `x` از نوع اعداد اعشاری `double`**
- **نوع مقدار بازگشتی تابع: عدد اعشاری `double`**

```
double f ( double x ) {
  return x/4 + cos(x);
}
```

- برای رسم منحنی توابع، نیاز به دستورات گرافیکی مثل رسم نقطه و پاره‌خط داریم که متأسفانه چنین دستوراتی به طور استاندارد در زبان C/C++ وجود ندارند.
- با این حال، برنامه‌هایی که منحنی یک تابع ریاضی را ترسیم می‌کنند درواقع تابع را به ازای تعداد زیادی از مقادیر متغیر که بر روی محور x فاصله‌ی کمی از یکدیگر دارند محاسبه می‌کنند و نقاط بدست آمده را با پاره خط‌های کوتاه به یکدیگر متصل می‌کنند.
- به جای ترسیم پاره‌خط و به عنوان تمرین، برنامه‌ای بنویسید که مختصات نقاط واقع بر منحنی تابع را چاپ کند.

```
#include <iostream>
#include <cmath>
using namespace std;
double f ( double x ) {
  return x/4 + cos(x);
}
```

```
int main(){
  double x;
  for(x=0;x<=10;x+=0.01) {
    double y = f(x);
    cout << "(" << x << "," << y << ")"
         << endl;
  }
  return 0;
}
```



```
#include <iostream>
#include <cmath>
using namespace std;

double f ( double x ) {
   return x/4 + cos(x);
}
int main(){
   double x;
   for(x=0;x<=10;x+=0.0
      double y = f(x);
      cout << "(" << x << "," << y << ")"
           << endl;
   }
   return 0;
}
```

اعلان تابع

بدنه‌ی تابع

فراخوانی تابع

شناسه‌ی تابع
Function identifier

- کلمه‌ای که برای نامگذاری تابع انتخاب می‌کنید باید با قواعد تعریف شناسه در زبان C/C++ موافق باشد؛ یعنی نباید از کلمات کلیدی زبان C/C++ باشد و نباید در حوزه‌ی تعریف آن، شناسه‌ی همنام دیگری وجود داشته باشد.

- همانند متغیرها، شناسه‌ی یک تابع می‌تواند ترکیبی از حروف الفبای انگلیسی و ارقام باشد ولی نباید با رقم شروع شود.
به عنوان مثال، `f`، `log10` و `str2num` می‌توانند برای نامگذاری یک تابع مورد استفاده قرار گیرند. اما تعریف تابعی با نام `double` در برنامه‌ی زبان C/C++ مجاز نیست.

آرگومان‌های تابع
Function arguments

- متغیری که به عنوان آرگومان (پارامتر) در اعلان تابع بکار گرفته می‌شود را اصطلاحاً پارامتر صوری (formal parameter) می‌گوئیم.
- مقدار ثابت، متغیر یا عبارتی که تابع به ازای آن فراخوانی می‌شود را اصطلاحاً پارامتر واقعی (actual parameter) می‌گوئیم.

```
#include <iostream>
#include <cmath>
using namespace std;

double f ( double x ) {
  return x/4 + cos(x);
}
int main(){
  double x;
  for(x=0;x<=10;x+=
    double y = f(x);
    cout << "(" << x << "," << y << ")"
         << endl;
  }
  return 0;
}
```

پارامتر صوری

پارامتر واقعی

ضرورتاً لازم نیست نام پارامتر صوری و پارامتر واقعی یکسان باشد.

```
double f ( double x ) {
  return x/4 + cos(x);
}

int main(){
  double t;
  for(t=0;t<=10;t+=0.01) {
    double y = f(t);
    cout << "(" << t << "," << y << ")"
         << endl;
  }
  ....
```

مثال ۲. تابع فاکتوریل.

- شناسه‌ی تابع: `fact`
- آرگومان تابع: `n` از نوع عدد صحیح `int`
- نوع مقدار بازگشتی: عدد صحیح `int`

```
int fact ( int n ) {
  int f = 1;
  int i;
  for ( i = 1; i<=n; i++ )
    f * = i;
  return f;
}
```

متغیرهای محلی
Local variables

- اصطلاحاً به متغیرهایی که در بدنه‌ی تعریف یک تابع، تعریف می‌شوند متغیر محلی یا متغیر موضعی می‌گویند.

```
int fact ( int n ) {
   int f = 1;
   int i;
   for ( i = 1; i<=n; i++ )
      f * = i;
   return f;
}
```

متغیرهای i و f متغیرهای محلی تابع fact هستند.

متغیرهای محلی Local variables

```
int fact ( int n ) {
  int f = 1;
  int i;
  for( i=1; i<=n; i++ )
    f * = i;
  return f;
}
int main ( ) {
  double f;
  int i = 5;
  cout << fact(i);
  ....
}
```

حوزه‌ی اعتبار متغیرهای محلی و امکان دسترسی به آن‌ها، تنها به بدنه‌ی تعریف همان تابع محدود است.

از این رو، بین متغیرهای محلی همنام در توابع مختلف، هیچ نوع تعارضی پدید نمی‌آید.

استفاده از توابع جدید تعریف شده در تعریف دیگر توابع

مثال ۳. تابعی برای محاسبه‌ی $\binom{n}{k}$.

- شناسه‌ی تابع: `nchoosek`
- آرگومان تابع: `n` و `k` هر دو از نوع عدد صحیح `int`
- نوع مقدار بازگشتی: عدد صحیح `int`

```
int nchoosek ( int n, int k ) {
  return fact(n)/(fact(k)*fact(n-k));
}
```

- برای فراخوانی تابع `fact` از داخل تابع `nchoosek` لازم است تابع `fact` پیش از تابع `nchoosek`، تعریف شده باشد.

مثال ۴. تابعی برای محاسبه‌ی ماکزیمم دو عدد صحیح.

- شناسه‌ی تابع: `max`
- آرگومان تابع: `x` و `y` هر دو از نوع `int`
- نوع مقدار بازگشتی: عدد صحیح `int`

راه حل ۱

```
int max ( int x, int y ) {
  if ( x > y )
    return x;
  else
    return y;
}
```

مثال ۴. تابعی برای محاسبه‌ی ماکزیمم دو عدد صحیح.

- فراموش نکنید که روال اجرای برنامه با رسیدن به دستور `return`، از تابع به برنامه‌ی فراخواننده جهش می‌کند و به نقطه‌ای که تابع در آن فراخوانی شده است بازمی‌گردد.
- البته، ضمن این کار مقدار بازگشتی تابع را نیز به برنامه‌ی اصلی می‌برد.

راه حل ۲

```
int max ( int x, int y ) {
  if ( x > y )
    return x;
  return y;
}
```

مثال ۴. تابعی برای محاسبه‌ی ماکزیمم دو عدد صحیح.

- با این حال بهتر است تنها از یک دستور **return** در تابع استفاده کنیم آن هم در آخرین خط از دستورات داخل تابع.
- این شیوه‌ای مطمئن‌تر برای برنامه‌نویسی توابع، بدون اشتباهات ناخواسته در پیاده‌سازی الگوریتم است.

راه حل ۳

```
int max ( int x, int y ) {
  int r;
  if ( x > y )
    r = x;
  else
    r = y;
  return r;
}
```

مقدار بازگشتی
Return value

- توابع بر حسب اینکه مقداری را (به عنوان نتیجه‌ی محاسبات) به برنامه‌ی فراخواننده گزارش می‌کنند یا نه، به دو دسته تقسیم می‌شوند.
- در اعلان توابعی که مقدار بازگشتی دارند نوع مقدار بازگشتی می‌بایست به طور صریح پیش از شناسه‌ی تابع نوشته شود و علاوه بر آن دست کم یک دستور `return` در روال اجرای دستورات درون بدنه‌ی تابع وجود داشته باشد.
- نوع داده‌ای مقداری که توسط دستور `return` گزارش می‌شود یا باید با نوع مقدار بازگشتی تصریح شده در اعلان تابع برابر باشد یا قابل تبدیل به آن باشد.

مثال ۵. تابعی که شکل بزرگ حرف داده شده را گزارش کند.

```
char upperCase ( char c ) {
  return c + 'A' – 'a';
}
```

اگر c از حروف کوچک الفبای انگلیسی نباشد کاراکتری بیربط گزارش می‌شود.

```
char upperCase ( char c ) {
  if ( 'a'<=c && c<='z' )
    return c + 'A' – 'a';
}
```

اگر c از حروف کوچک الفبای انگلیسی نباشد هیچ مقداری گزارش نمی‌شود =>

```
char upperCase ( char c ) {
  if ( 'a'<=c && c<='z' )
    c = c + 'A' – 'a';
  return c;
}
```

تابع بدون مقدار بازگشتی
void function

- برای بسیاری از توابع، آنچه مهم است عملیاتی است که با فراخوانی تابع به انجام می‌رسد و منطقاً تابع بدون مقدار بازگشتی است.
- با استفاده از کلمه‌ی کلیدی `void` به جای نوع مقدار بازگشتی می‌توانید توابعی بدون مقدار بازگشتی در زبان C/C++ تعریف کنید.

تابع بدون مقدار بازگشتی

void function

مثال ۶. تابعی که دو عدد داده شده را به شکل زوج مرتب چاپ می‌کند.

```
void printPair ( int a, int b ) {
  cout << '(' << a
       << ',' << b << ')';
}
```

تابع بدون مقدار بازگشتی
void function

مثال ۷. تابعی که تاریخ را به صورت روز/ماه/سال چاپ کند.

```
void printDate( int d, int m, int y )
{
  cout << y << '/' << m << '/' << d;
}
```

مثال ۷. تابعی که تاریخ را به صورت روز#ماه#سال چاپ کند.

فرض کنید علامت جداکننده (#)، نیز به عنوان پارامتر به تابع داده می‌شود.

```
void printDate( int d, int m, int y,
                char delim )
{
  cout << y << delim
       << m << delim << d;
}
```

نمونه‌هایی از فراخوانی این تابع.

```
printDate(29,12,1389,'/');
printDate(31,12,2010,'-');
```

Function arguments

آرگومان‌های تابع

- در اعلان تابع، نوع داده‌ای باید برای هر یک از آرگومان‌های تابع به طور جداگانه تصریح شود.

  به عنوان مثال، تعریف زیر از نظر نگارشی نادرست است.

```
void printDate( int d, m, y, char delim )
```

- در فراخوانی تابع، تعداد، ترتیب و نوع پارامترهای واقعی باید مطابق با تعداد، ترتیب و نوع پارامترهای صوری در اعلان تابع باشد.

  به عنوان مثال، فراخوانی‌های زیر نادرست‌اند.

```
printDate(29,12);
printDate(1389,12,29,'/');
printDate(31,12,2010,"-");
```

توابع بدون پارامتر
parameterless functions

مثال ۸. تابعی که توضیحاتی درباره‌ی برنامه چاپ می‌کند.

```
void printAbout( ) {
  cout << "MasterMind version 2.0" << endl;
  cout << endl;
  cout << "(C) 2010, John Smith" << endl;
  cout << "Email bug reports to
johnsmith@abc.com" << endl;
  cout << endl;
}
```

توابع بدون پارامتر
parameterless functions

- در زبان C/C++، برای فراخوانی توابع می‌بایست لیست پارامترها پس از نام تابع، بین علامت (و) نوشته شود. این قاعده حتی برای فراخوانی توابع بدون پارامتر نیز باید رعایت شود؛ یعنی یک جفت علامت (و) خالی جلوی نام تابع نوشته شود.

```
int main( ) {
   ....
   printAbout;    // Error
   printAbout(); // OK
   ....
}
```

توابع بدون پارامتر
parameterless functions

`int rand( );`	تابع مولد اعداد تصادفی.

- این تابع هیچ پارامتر ورودی ندارد اما با هر فراخوانی آن، یک عدد صحیح تصادفی گزارش می‌شود.
- محدوده‌ی اعداد تولید شده توسط این تابع از 0 تا `RAND_MAX-1` است.
- تابع `rand( )` و ثابت `RAND_MAX` در کتابخانه‌ی `cstdlib` تعریف شده‌اند.

درباره‌ی تابع مولد اعداد تصادفی

- کامپیوترهایی که امروزه ما از آن‌ها استفاده می‌کنیم همچنان در رده‌ای از ماشین‌های محاسبه قرار می‌گیرند که تمامی محاسبات در آن‌ها به صورت کاملاً تعیّنی (deterministic) انجام می‌شود.
- از این رو، جای سئوال است که چگونه ممکن است بتوان تابعی داشت که هر فراخوانی آن، نتیجه‌ی متفاوتی بدست آورد.
- در واقع، برنامه‌ای که به مفهوم دقیق ریاضی، اعداد تصادفی تولید کند نمی‌تواند بر روی این کامپیوترها وجود داشته باشد.

درباره‌ی تابع مولد اعداد تصادفی

- توابعی مثل تابع rand( ) در زبان C، دنباله‌ای از اعداد تولید می‌کنند که با اینکه در ظاهر تصادفی به نظر می‌رسند اما در واقع، دنباله‌ای قانونمند از اعدادند. چنین اعدادی را اصطلاحاً اعداد شبه تصادفی و تابع مولد آن‌ها را تابع مولد اعداد شبه تصادفی نامیده می‌شوند.

```
int main( ) {
  while(true)
    cout << rand() << "   ";
  return 0;
}
```

درباره‌ی تابع مولد اعداد تصادفی

File Edit View Scrollback Bookmarks Settings Help

```
087  42394625  780624169  1056975347  688189004  1839443114  264840222  539494730  1819419794  716470637  12221948
12  1638648563  766968804  1080642466  635278780  167662703  715698178  2134891624  433549979  759728983  11042048
379  727862520  1047032224  275499839  387931197  160269085  696610379  578528229  358111464  664767819  620922854
3  1721743166  1309111858  830695099  1986583389  1848606588  502631245  555570378  1970826072  69219210  46735294
  1149861676  682014074  757973932  1865559854  669422051  1191523911  477805189  1773626875  837830642  120566770
1  1113330482  1593598906  833444537  1809940861  24643487  1191556001  327225032  645566342  182807987  204896819
00  1013503086  1888067940  1655801141  1516134332  296154670  1479143565  1585353542  342889964  2069454793  5877
904039  679945077  305807777  1694326090  1871468989  783612966  1320469317  561815983  1989280676  1993644769  16
435395934  679605658  1337603679  1460039422  1871161659  1664828711  2105605764  2053969646  1566313262  19128003
85  1306897554  1421117809  288639769  1603052225  752777726  1873993311  1945942189  674748872  314241233  823362
3949  620049010  370205022  1078679290  1403661977  1690674340  1640495274  1245459005  1536835461  1168158091  53
957471  358278122  1993410713  1940119130  2023106834  1951532829  1846605129  1441936448  1716849498  619110566
990483659  907750335  56918932  1743261386  634259998  2002861121  270526610  948501231  678740054  1625220559  15
048945076  556416202  824728571  592135768  49427828  2070187576  2128971229  1217585919  456075219  50445052  157
2002285  1990564183  1451487228  106051466  1689685664  745940028  1822900964  161312582  1347290383  665900976  1
1404209315  261678714  1703322915  1259586789  532205324  504340498  1938326843  9942235  2072890740  839788271  5
50135663  1431924040  615786265  672839591  1413411621  1833372185  1128914811  1463856674  1261752579  1430917096
9  565756159  1536968562  849139225  1311696187  1212385879  1010451807  511502923  1878286855  2079514724  191571
65569  1635353991  1027815379  524687245  2139694489  818658574  534629480  2065101582  1658446846  1100987918  66
2887238  1716774183  1340593189  208815211  1402662720  322024352  1672671885  516931651  1752941448  832125446  1
1142426362  1681264671  246900350  207328593  544232830  758403273  2085615448  476263906  526631863  2078097369
 1554447243  455300966  2103828739  225622169  989930447  2021446673  1884069015  2090918365  541716622  679472605
0  1882309811  888287817  915387973  56850515  413476054  1432319624  1809791963  1245601501  367523787  804734678
  614424137  1012063271  1323615355  1372827410  950195072  1799879261  1899459273  880808793  1764013511  1306422
9760  1720358602  1532045038  178556559  1594321627  1268630405  121991276  2136038249  1948103011  1782200176  18
88907180  550104501  1927714928  1102383234  1982424126  1590023244  200501087  202464265  247274274  979883612  8
259337545  156015319  42232164  62048969  1955894580  1941691437  942857763  1572424443  1100630658  131483875  11
85192048  310040434  592137376  1753822453  432031710  580691978  1554441816  66748238  304072743  95865348  61685
023  1198248583  451793218  1674327267  1398749670  654257483  1921601541  231149634  1471145885  1033455439  3871
378049  1095504408  195575886  1307585838  2038362171  1768000329  260732848  22362398  765816079  745924896  3324
953455  352263702  764434542  1938645433  1906705518  831182781  95234528  2002570867  1448035521  179538552  1053
```

workspace : bash

درباره‌ی تابع مولد اعداد تصادفی

مثال ۹. تابعی که اعداد اعشاری تصادفی در بازه‌ی [0,1) تولید کند.

```
double random( ) {
  return double(rand())/RAND_MAX;
}
```

درباره‌ی تابع مولد اعداد تصادفی

مثال ۱۰. تابعی که پرتاب‌های یک تاس را شبیه‌سازی می‌کند؛ یعنی هر فراخوانی آن، یک عدد از بین اعداد صحیح ۱ تا ۶ را به طور تصادفی گزارش می‌کند.

راه حل ۱

```
int toss( ) {
  return 6*random()+1;
}
```

راه حل ۲

```
int toss( ) {
  return rand()%6 + 1;
}
```

درباره‌ی تابع مولد اعداد تصادفی

مثال ۱۱. تابعی که با هر فراخوانی آن، یک عدد از بین اعداد صحیح a تا b به طور تصادفی گزارش می‌شود.

راه حل ۱

```
int myRand( int a, int b ) {
  return (b-a+1)*random()+a;
}
```

راه حل ۲

```
int myRand( int a, int b ) {
  return rand()%(b-a+1) + a;
}
```

درباره‌ی تابع مولد اعداد تصادفی

مثال ۱۲. تابعی که با هر فراخوانی آن، یک عدد اعشاری از بازه‌ی [a,b) به طور تصادفی گزارش می‌شود.

```
double urand( double a, double b ) {
  return (b-a)*random()+a;
}
```