

برنامه سازی با کامپیوتر



اطلاعات – Information

برای اینکه از **داده ها** بتوانیم بهتر استفاده کنیم لازم است بر روی آنها محاسبات و یا به طور کلی عملیاتی را انجام دهیم. نتایج حاصل از این عملیات را **اطلاعات** می نامیم که می تواند مورد تفسیر و بررسی قرار

گیرد



ارتباط بین داده‌ها، عملیات و اطلاعات

پردازش – Process

مجموعه محاسبات و عملیاتی که بر روی داده ها صورت می گیرد را **پردازش** می نامند. پردازش، گاهی ساده مانند محاسبه مجموع و یا خارج قسمت دو عدد می باشد و یا گاهی پیچیده مانند تشخیص شماره پلاک خودرو با استفاده از عکس گرفته شده از خودرو، توسط یک کامپیوتر است.

دانش

نتیجه بررسی اطلاعات بدست آمده به دانش ختم گردد که دانش می تواند مبنای تصمیم گیری برای انجام کاری شود.

توجه به نکاتی کلیدی در پردازش

• صحت داده ها

- یعنی داده ها به درستی گردآوری شده و داده اشتباه بین آنها نباشد.

• درستی انجام محاسبات

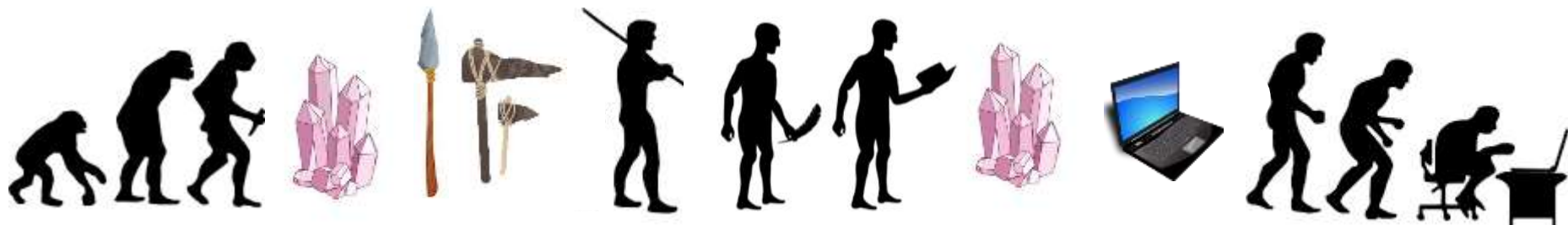
- محاسبات و یا عملیاتی که بر روی داده ها صورت می گیرد با دقت و بدون اشتباه انجام شوند.

• روش انجام پردازش

- باید پردازش مناسب روی آنها انجام دهیم تا به اطلاعات مفید برسیم (استفاده از روش های بهینه و الگوریتم های مناسب)

معمولاً از کامپیوتر برای انجام پردازش بر روی داده ها استفاده می کنیم چون سرعت کامپیوتر در اجرای عملیات و دقت انجام محاسبات بالا است. همچنین از کامپیوتر برای ذخیره و نگهداری

داده ها استفاده می شود تا بعداً پردازش بر روی آنها صورت گیرد.



- مغز کامپیوتر از سنگ ساخته شده است، بنابراین از جمودات است. حال آنکه امروزه کامپیوترها تا حدی پیشرفت کرده اند که حتی گاهی با یک موجود زنده و یا حتی با یک انسان کاملاً هوشمند ممکن است اشتباه گرفته شوند. تمامی قسمت های فیزیکی سخت یا جامد کامپیوترها را اصطلاحاً سخت افزار (Hardware) می نامند.

- اما آنچه که به این اجزای بی جان، پویایی و روح می بخشد، اجرای برنامه ها و سلسله دستوراتی است که انسان ها برای آنها تعیین کرده اند و اصطلاحاً به آن **نرم افزار (Software)** گفته می شود.

- این مخلوق دست بشر برخی ویژگی های فرا بشری دارد که مهمترین آنها عبارتند از:

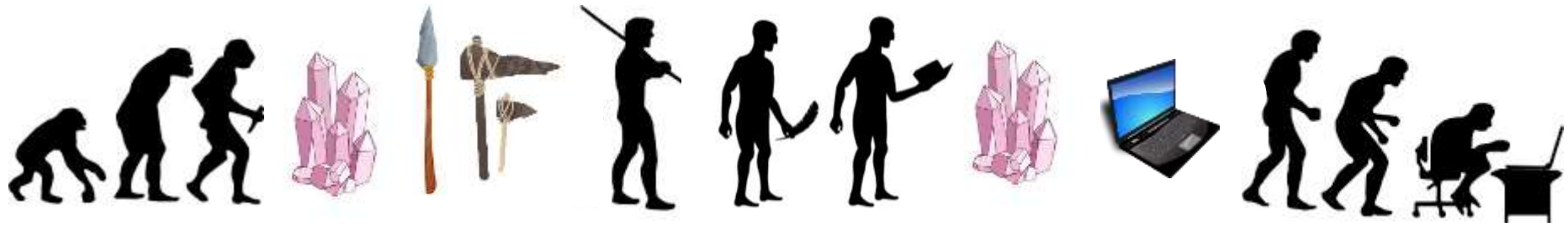
- از تکرار خسته نمی شود

- اشتباه نمی کند

- فراموش نمی کند

- تحت تاثیر عواطف و یا عوامل بیرونی (تا جایی که دچار مشکل سخت افزاری نشود) قرار نمی گیرد

- از لحاظ دقت و سرعت اجرا در امور بسیار فراتر از قدرت بشر است



• نرم افزار ها چندین رده دارند:

(۱) فریمور (Firmware) یا میان افزار یا نرم افزار سطح پایین

BIOS (بایوس) که مخفف Basic Input/Output System

UEFI که مخفف Unified Extensible Firmware Interface

این دو وظیفه راه اندازی اولیه و تست سلامت قطعات سخت افزاری ضروری سیستم را به عهده دارند و پس از آن که مطمئن شدند همه قطعات سالم هستند و مشکلی ندارند، سیستم عامل را اجرا کرده و کنترل سیستم را به او می سپارد. تنظیمات bios یا uefi در حافظه ای نگهداری می شود که انرژی خود را از طریق باتری ای که بر روی مادربرد وجود دارد تامین می کند.

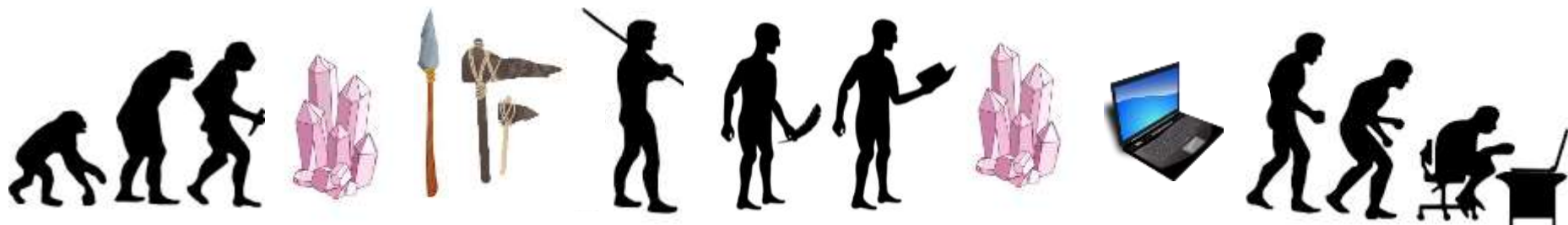
(۲) سیستم عامل (operating system) یا نرم افزار سطح بالا

(۳) نرم افزار کاربردی (Application) یا برنامه (Program). از آنجایی که اغلب با این رده از نرم افزارها سروکار دارند اصطلاحاً به اپلیکیشن ها نرم افزار (Software) می گویند.

سوال؟

**با وجود این همه نرم افزار مفید
و آماده آیا باز هم لازم است
برنامه نویسی را یاد بگیریم؟**





کامپیوتر آمده است تا نیازهای انسان را رفع کند و در این راستا از برنامه ها یا اپلیکیشن ها استفاده می شود.

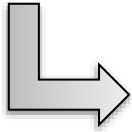
نیازهای انسانها دو دسته اند: (الف) نیازهای عام (ب) نیازهای خاص

از آنجایی که نیازهای عام، شامل بسیاری از افراد جامعه می شود، بنابراین ارائه راه های بهتر برای رفع این نیازها همواره توجیه اقتصادی دارد. بنابراین برنامه نویسان همواره سعی می کنند تا این نیازهای عام را شناسایی کرده و نرم افزارهای مرتبط با آنها را توسعه دهند. انبوه بسته های نرم افزاری موجود از قبیل Office، Media Player، Photoshop، Calculator و غیره تماماً برای این منظور ارائه شده اند.

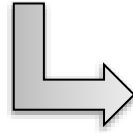
اما برخی نیازهای خاص وجود دارند که عمومیت نداشته و از آنجایی که متقاضی کمی دارد، بنابراین توسعه نرم افزار خاص برای رفع آن نیاز، توجیه اقتصادی ندارد. پس انسانها برای رفع این گونه نیازها یا باید تمام هزینه های توسعه نرم افزار مورد نیاز را تقبل کنند، یا اینکه روش های برنامه سازی را آموخته و برای خود آن برنامه ای که می خواهند طراحی کنند و بسازند.



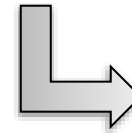
تعریف نیاز



برنامه نویسی



تفہیم به ماشین



ایجاد برنامه و رفع نیاز

برنامه – Program

به مجموعه دستوراتی که به کامپیوتر می فهماند که چه نوع پردازشی را بر روی داده ها انجام دهد و همچنین اطلاعات به دست آمده را چگونه نمایش دهد **برنامه** می گویند.

زبان قابل فهم کامپیوتر دنباله ای از کدهای صفر و یک است که به آن **زبان ماشین** می گویند.

پس از اینکه برنامه با زبان دیگری غیر از زبان ماشین نوشته شد، با استفاده از یک **کامپایلر (هم گردان یا مترجم)**، باید به زبان ماشین تبدیل شود تا کامپیوتر بتواند آن را بفهمد و اجرا نماید.

کامپایلر خود نیز یک برنامه کامپیوتری می باشد که وظیفه آن، ترجمه و تبدیل دستورات یک زبان سطح بالا، به کدهای زبان ماشین می باشد.

برنامه نویسی

نوشتن دستورات لازم برای کنترل نحوه کار کامپیوتر، به طوری که کامپیوتر بتواند یک کارمشخص را انجام دهد را **برنامه نویسی** میگویند. **برنامه نویس** شخصی است که آشنا به دستورات یک زبان برنامه نویسی باشد و با به کارگیری صحیح و مناسب دستورات، برنامه نویسی کند.



انواع زبان های برنامه نویسی

عموماً زبان های برنامه نویسی را به پنج نسل تقسیم می کنند:

نسل اول زبان ماشین یا زبان صفرو یک است.

نسل دوم زبان هایی مانند اسمبلی که قابل فهم تر برای انسان است

نسل سوم زبان هایی که دستورهای آن قابل فهم تر برای انسان بوده و برای فهم ماشین نیاز به کامپایلرها دارند.

نسل چهارم مثل زبان برنامه نویسی متلب بسیار نزدیک تر به زبان محاوره های انسانی است.

نسل پنجم به گروهی از زبان های برنامه نویسی گفته می شود که از یک **GUI** به معنی «رابط گرافیکی کاربری» برای کدنویسی استفاده می شود که در نهایت کد نوشته شده با این زبان ها، با کامپایلر یک زبان **GL ۳** یا **GL ۴** کامپایل می شود.

انواع زبان های برنامه نویسی

۱- زبان های سطح پایین (Low Level)

زبان هایی که به زبان پردازشگر کامپیوتر (CPU) نزدیک بوده و مسلماً از زبان محاوره ای ما دور هستند. همانند: زبان ماشین و زبان اسمبلی.

Example

۲- زبان سطح میانی (Medium Level)

در این زبان ها برخی دستورات وجود دارد که مستقیماً به سخت افزار ارتباط پیدا کرده و همچنین برخی دستورات نیز وجود دارد که به زبان محاوره انسان نزدیک است. همانند زبان C. مناسب برای برنامه ریزی کردن ریز پردازنده ها (Micro Processors).



۳- زبان سطح بالا (High Level)

زبان هایی که به زبان محاوره ما نزدیک باشند. همانند ویژوال بیسیک و سی شارپ

Microsoft
Visual Basic



انواع زبان های برنامه نویسی

۱- زبان ماشین (Machine Code)

فهم دستورات زبان ماشین و همچنین خطایابی در آن بسیار دشوار بود. به همین دلیل زبان اسمبلی بوجود آمد.

۲- زبان اسمبلی (assembly code)

به دلیل کاربرد حروف به جای اعداد در این زبان، کار با آن راحت تر است. به مبدل حروف به اعداد برای فهم کامپیوتر اسمبلر گفته می شود.

۳- زبان سطح بالا (High Level Code)

به دلیل کند بودن زبان اسمبلی، زبان سطح بالا جایگزین آن شد. در زبان سطح بالا از کلمات و علائم و نشانه های محاوره ای به سادگی استفاده می شود و کار با آن آسان تر است.

```
0000: 00 00 48-20 43 48 46-43 46 44 45  H CKFDENECD E
0010: 46 46 43 46-47 45 46 46-41 43 41 43  FFCFGEFFCCACAC
0020: 41 43 41 43-41 00 20 45-44 45 42 45  ACACA EMEPEDEBE
0030: 4D 45 49 45-50 46 44 46-41 43 41 43  METEPPDFECACAC
0040: 41 43 41 43-41 41 41 00-00 00 00 7A  ACACAA z
0050: FF 53 4D 42-72 00 00 00-00 00 00 00  SMGr TS
0060: 00 00 00 00-00 00 00 00-00 00 00 00  9
0070: 00 57 00 02-50 43 20 4E-52 4B 20 50  W PC NETWORK P
0080: 52 4F 47 52-41 4D 20 31-4C 41 4E 4D  ROGRAM 1 0 LANM
0090: 41 4E 31 2E-30 00 02 57-77 73 20 66  ANL 0 Windows f
00A0: 6F 72 20 57-6F 72 6B 67-73 20 33 2E  or Workgroups 3.
00B0: 31 61 00 02-4C 4D 31 2E-32 00 02 4E  1a 0LMI 2X002 0N
00C0: 54 20 4C 4D-20 30 2E 31-10 8F FF 53  T LM 0.12 S
00D0: 4D 42 73 00-00 00 00 18-00 00 00 00  MBs: 1 9
00E0: 00 00 00 00-00 00 00 00-00 00 0C FF  9
00F0: 00 00 00 04-11 0A 00 00-00 7E 10 00  9
0100: 00 00 00 04-00 00 80 7E-7A 06 06 2B  H B z
0110: 06 01 05 05-02 00 82 10-6A 01 82 10  B B j B
0120: 66 23 82 10-62 03 82 04-41 41 41 41  F B B B A A A A A
0130: 41 41 41 41-41 41 41 41-41 41 41 41  A A A A A A A A A A A
0140: 41 41 41 41-41 41 41 41-41 41 41 41  A A A A A A A A A A A
```

```
FE30
*FDEE
FDEE- 6C 36 00 JMP ($0036)
FDEE 6C 36 00 CMP #A0
FDEE 6C 36 00 BCC #FDF6
FDEE 6C 36 00 AND #32
FDEE 6C 36 00 STY #35
FDEE 6C 36 00 PHA
FDEE 6C 36 00 JSR #FB78
FDEE 6C 36 00 PLA
FDEE 6C 36 00 LDY #35
FDEE 6C 36 00 RTS
FDEE 6C 36 00 DEC #34
FDEE 6C 36 00 BEQ #FDA3
FDEE 6C 36 00 DEC #34
FDEE 6C 36 00 BNE #FE1D
FDEE 6C 36 00 CMP #BA
FDEE 6C 36 00 BNE #FDC6
FDEE 6C 36 00 STY #31
```

```
1 import java.util.Scanner;
2
3 public class Hello
4 {
5     public static void main(String[] args)
6     {
7         System.out.print("Please enter your name: ");
8         String name = new Scanner(System.in).next();
9
10        System.out.println("Hi "+name+"!");
11    }
12 }
```

انواع زبان های برنامه نویسی

کد سطح بالا

High Level Code

```
i = 1; j = 1;
while (true){
*val++ = i+j;
j = i+(i=j);}
```

```
mov r0,#1
mov r1,#1
l:
add r2,r0,r1
str r2,[r3]
add r3,#4
mov r0,r1
mov r1,r2
b l
```

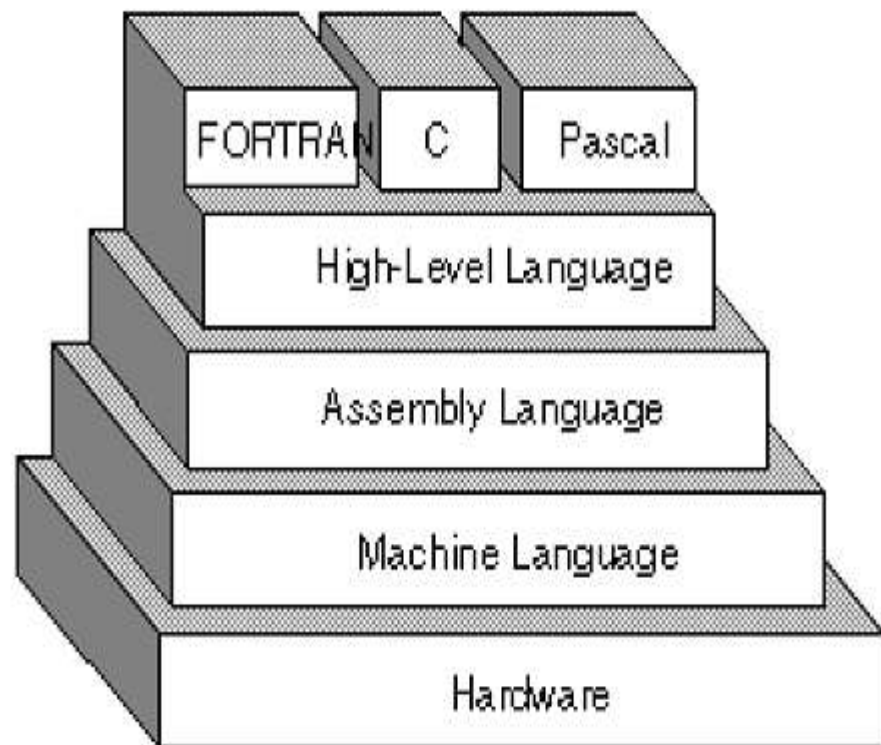
Assembly Code

```
11010011110100000
00000000000000001
11010011110100000
00010000000000001
11010000100000000
00100000000000000
11010101100000011
00100000000000000
11010010100000011
00110000000000100
11010001101000000
00000000000000001
11010001101000000
00010000000000010
10111010111111111
1111111111111001
```

Binary Code

کد اسمبلی

کد ماشین



دسته بندی معروف ترین زبان های برنامه نویسی

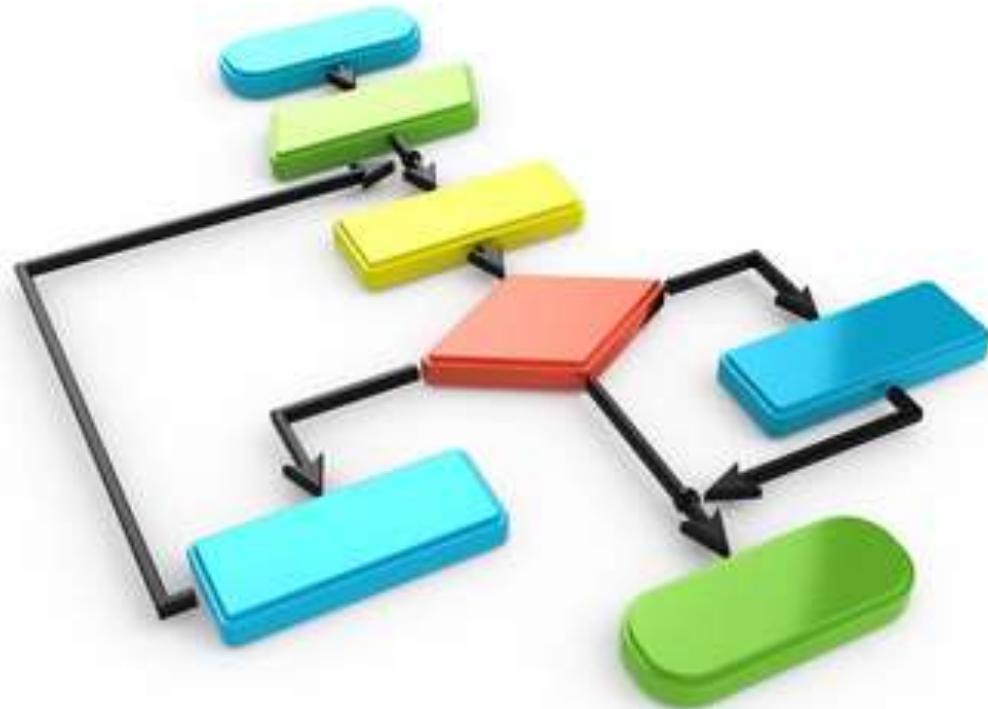
زبان سطح میانی: هم مانند زبان های سطح پایین مانند اسمبلی مستقیماً به حافظه دستیابی داشته و با مفاهیم بیت و بایت و آدرس کار می کند و هم مانند زبان های سطح بالا دستورالعمل های آن به به زبان طبیعی نزدیک تر بوده و قابلیت خوانایی و غلط یابی بالایی دارد.

زبان ساخت یافته: زبانی که دارای ساختارهایی همچون شرط و تکرار است که قابلیت خوانایی برنامه ها را بالا می برد و درک آن را آسان تر می کند.

زبان های سطح پائین	زبان های میانی	زبان های سطح بالا
ماکرواسمبلر اسمبلر	جاوا فورث C, (C++)	پاسکال ادا ماجولا-۲ کوبول بیسیک

زبان های ساخت یافته	زبان های غیر ساخت یافته
پاسکال ادا جاوا ماجولا-۲ C, (C++)	فرترن بیسیک کوبول

الگوریتم و فلوچارت



الگوریتم

- الگوریتم و فلوجارت تنها چیزهایی هستند که به طور کامل میان تمامی زبان های برنامه نویسی مشترک هستند.
- الگوریتم مجموعه دستورالعمل های مشخصی است که مراحل انجام یک کار و یا حل یک مسئله را با زبانی دقیق و با جزئیات کافی (اعم از ترتیب و توالی امور) ذکر می کند.
- کلمه الگوریتم (**Algorithmus**) از نام دانشمند بزرگ و پر آوازه یعنی محمد ابن موسی الخوارزمی (الخوریسم) برگرفته شده است.



الگوریتم

هر الگوریتم دارای سه بخش اصلی است:

آغاز - دستورالعمل ها - پایان

انواع مختلف دستورالعمل ها:

۱- دستورالعمل های محاسباتی و انتسابی: نسبت دادن مقادیر به متغیر ها یا اجرای محاسبات

۲- عبارات توضیحی: برای اضافه کردن توضیحات به برنامه یا الگوریتم کاربرد دارد و بایستی از دستورالعمل ها تفکیک شود.

۳- دستورالعمل های شرطی: در این دستورالعمل یک شرط از پیش تعیین شده بررسی می شود.

۴- دستورالعمل های خروجی: نتایج بدست آمده از محاسبات و تحلیل ها را چاپ می کند.

الگوریتم نویسی

برای حل یک مسئله در کامپیوتر باید آن را به مراحل مختلف شکست تا اینکه به صورت عملیات و تصمیم گیری های ساده ای که کامپیوتر قادر به درک آنها باشد تبدیل شوند.

یک الگوریتم بایستی این ویژگی ها را داشته باشد:

- ۱- در نهایت به نتیجه مورد نظر برسد.
- ۲- در زمان محدود و بهینه پایان یابد.
- ۳- دستورالعمل ها بایستی به ترتیب منطقی قرار گیرند.
- ۴- جملات الگوریتم بایستی امری و یا سوالی باشند.
- ۵- هر الگوریتم بایستی نقطه آغاز و پایان داشته باشد.

الگوریتم نویسی

الگوریتم تعویض چرخ پنجر شده یک ماشین:



- ۰- شروع.
- ۱- جک را زیر اتومبیل بگذارید.
- ۲- پیچهای چرخ پنجر شده را باز کنید.
- ۳- چرخ را خارج کنید.
- ۴- چرخ یدک را به جای چرخ پنجر شده بگذارید.
- ۵- پیچها را ببندید.
- ۶- اگر پیچها سفت نشده اند به مرحله ۵ برو.
- ۷- جک را پایین بیاورید.
- ۸- چرخ پنجر شده را در صندوق عقب اتومبیل بگذارید.
- ۹- پایان.

الگوریتم نویسی

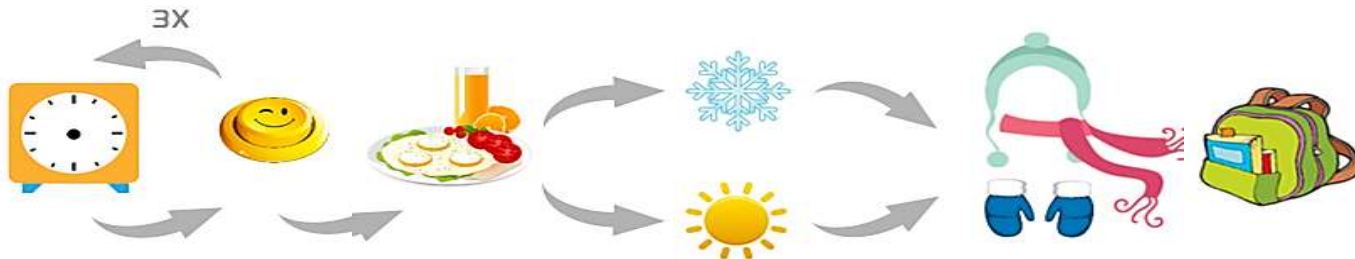
الگوریتم دریافت دو عدد از ورودی، محاسبه حاصل جمع آنها و بررسی بزرگتر یا کوچک تر بودن حاصل جمع از عدد ۲۰:

- ۰- شروع .
- ۱- دو عدد a و b را از ورودی دریافت کن.
- ۲- $a+b$ را محاسبه کن.
- ۳- آیا $a+b < 20$ است؟ اگر بلی به مرحله ۶ برو.
- ۴- بنویس خیر.
- ۵- به مرحله ۷ برو.
- ۶- بنویس بلی.
- ۷- پایان .

الگوریتم نویسی

هرگونه کم توجهی و بی دقتی در نوشتن الگوریتم، ضمن بروز مشکلات و اشتباهات، ممکن است برنامه ریزی را منحرف نموده و از هدف نهایی دورسازد.

الگوریتم نویسی اولین گام در برنامه نویسی بوده و درک صحیح و کامل صورت مسئله نیز اولین گام در الگوریتم نویسی است.



الگوریتم نویسی

مثال: الگوریتمی بنویسید که مجموع اعداد طبیعی مضرب ۷ و کوچکتر از ۵۰ را حساب کند.

- ۰- شروع.
- ۱- در خانه حافظه sum عدد صفر را قرار بده.
- ۲- در خانه حافظه index عدد ۷ را قرار بده.
- ۳- مقدار index را با مقدار sum جمع کن
و حاصل را در sum قرار بده.
- ۴- مقدار ۷ را با مقدار index جمع کن
و حاصل را در index قرار بده.
- ۵- آیا index بزرگتر از ۵۰ است، اگر خیر به مرحله ۳ برو.
- ۶- محتوای sum را چاپ کن.
- ۷- پایان.

الگوریتم نویسی

تمرین: الگوریتمی بنویسید که ۱۰۰۰ عدد را از ورودی دریافت کرده و کوچکترین را چاپ کند:

- ۰- شروع.
- ۱- \min را دریافت کن.
- ۲- $i = 1$.
- ۳- a را دریافت کن.
- ۴- اگر $\min > a$ آنگاه $\min = a$.
- ۵- $i = i + 1$.
- ۶- اگر $i \leq 1000$ به مرحله ۸ برو.
- ۷- به مرحله ۳ برو.
- ۸- \min را چاپ کن.
- ۹- پایان.

الگوریتم نویسی

تمرین: الگوریتمی بنویسید که سه عدد از ورودی دریافت شود و تعیین شود که این اعداد می توانند اضلاع یک مثلث باشند یا خیر:

- ۰ - شروع .
- ۱ - a و b و c را از ورودی بگیر .
- ۲ - اگر $b+c < a$ به ۷ برو .
- ۳ - اگر $a+c < b$ به ۷ برو .
- ۴ - اگر $a+b < c$ به ۷ برو .
- ۵ - بنویس " بلی " .
- ۶ - به ۸ برو .
- ۷ - بنویس " خیر " .
- ۸ - پایان .

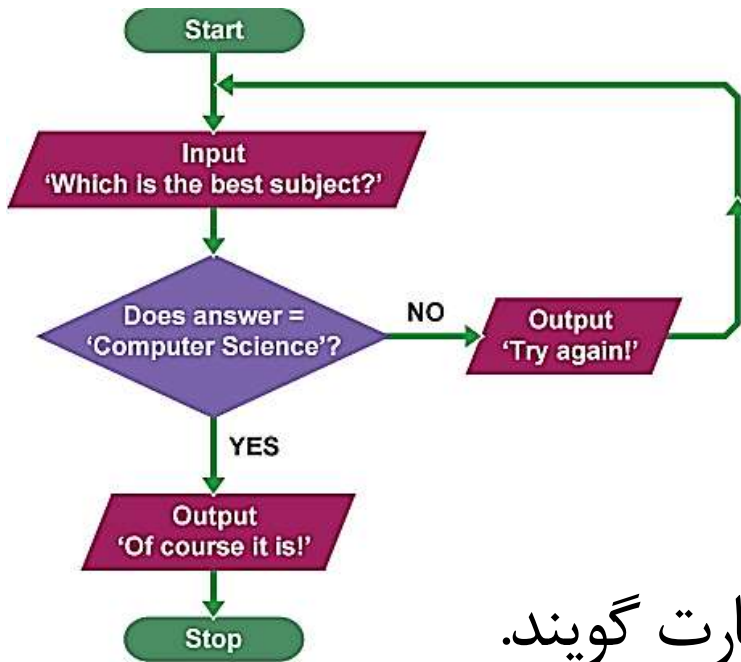
الگوریتم نویسی

چنانچه الگوریتم ها به زبان نوشتاری و با کلمات عامیانه نوشته شوند سبب طولانی شدن و حجیم شدن الگوریتم می شوند.

به منظور راحتی درک الگوریتم و همچنین سادگی نگارش آن از جملات محاوره ای ریاضی به همراه علائم قراردادی استفاده می شود.

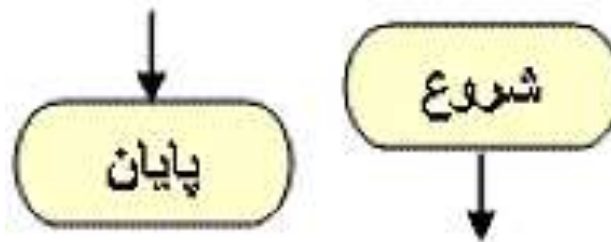


فلوچارت



روش تصویری و استاندارد نمایش الگوریتم را فلوچارت گویند. در رسم فلوچارت علائم و نمادهای استاندارد به کار می رود که هر کدام دارای معنای ویژه ای هستند.

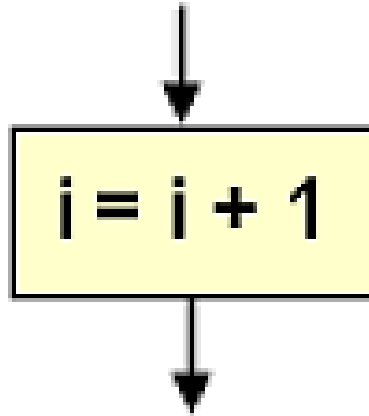
- از شکل بیضی خوابیده برای شروع و پایان عملیات استفاده می شود.



فلوچارت

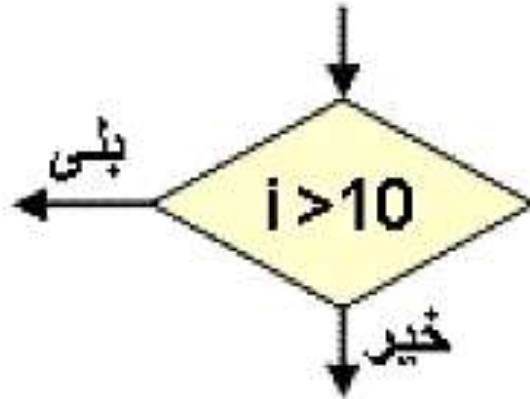
از مستطیل برای نمایش مراحل پردازشی استفاده می شود و در داخل آن عمل مورد نظر نوشته می شود.

این نماد می تواند چند ورودی داشته باشد ولی تنها یک خروجی دارد.



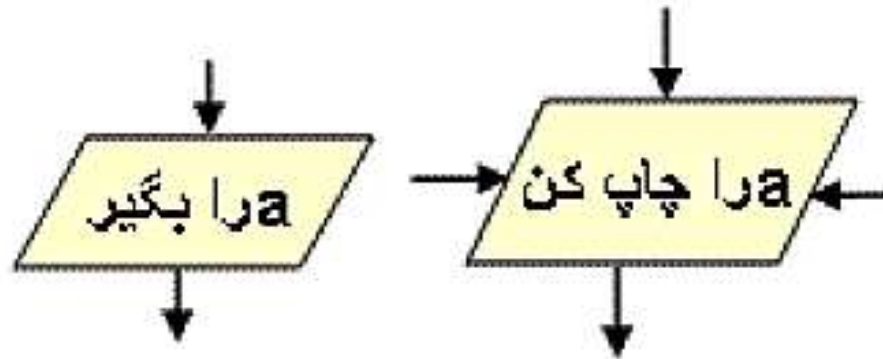
فلوچارت

از نماد لوزی برای نشان دادن مراحل تصمیم گیری استفاده می گردد و شرط یا سوال مورد نظر در داخل لوزی نوشته می شود.

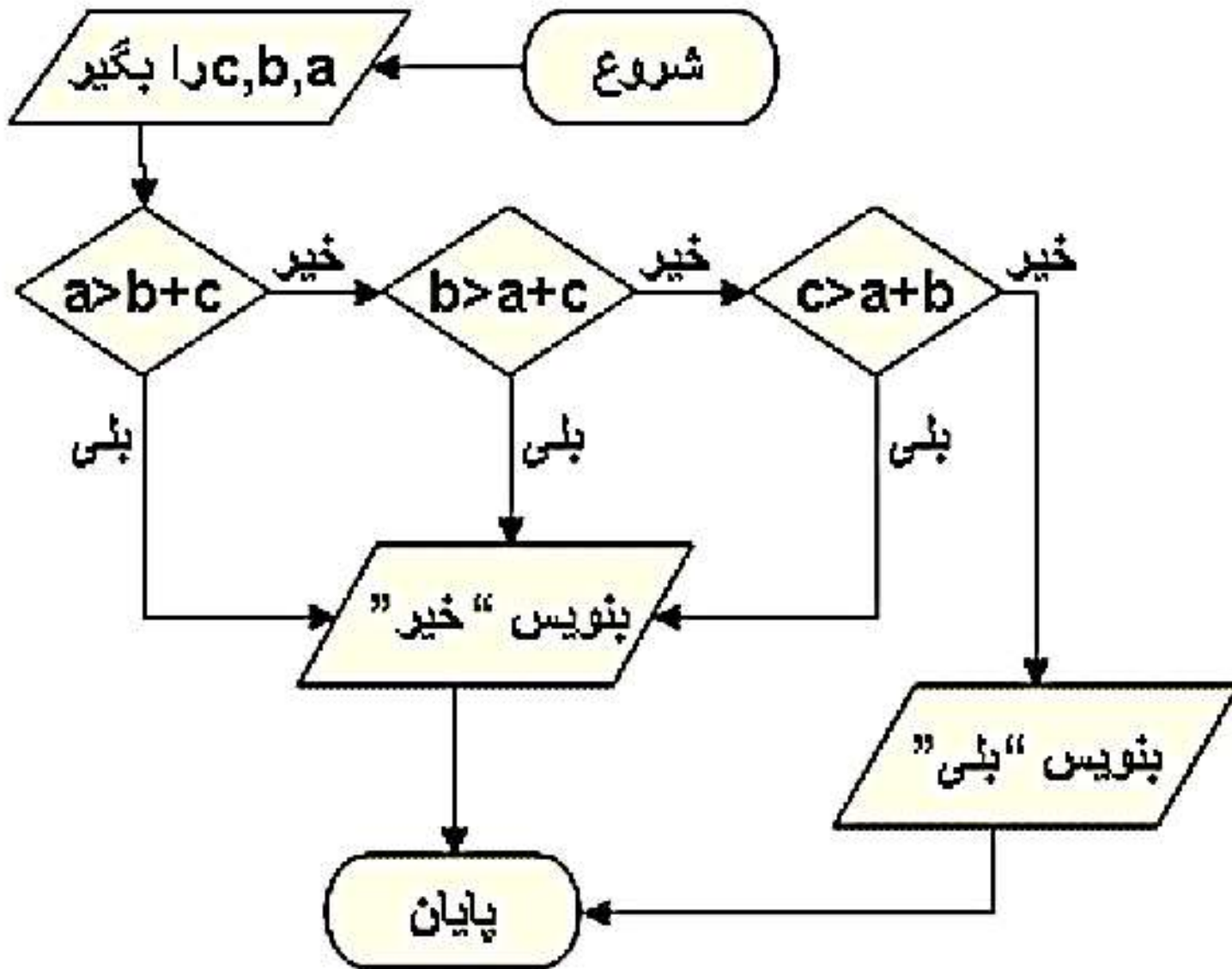


فلوچارت

از متوازی الاضلاع برای نشان دادن ورودی و خروجی استفاده می شود.

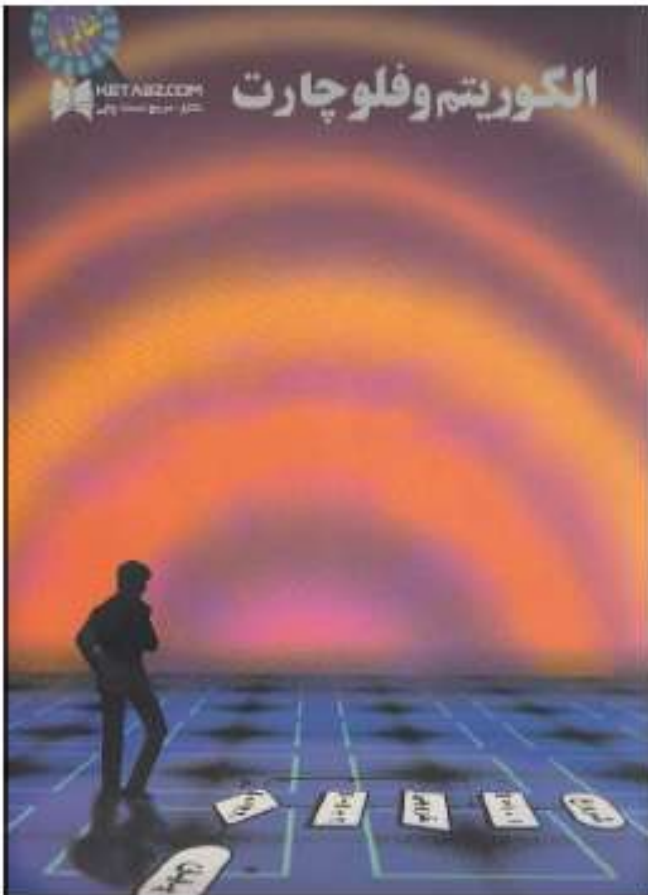


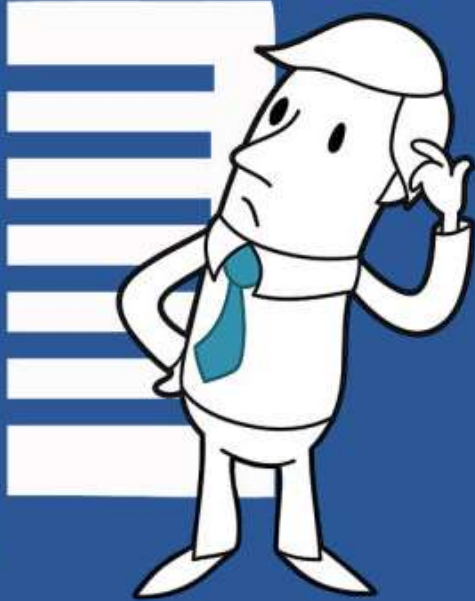
مثال: فلوچارت الگوریتم اضلاع مثلث در مثال قبل را بنویسید.



کتاب مرجع تکمیلی

- الگوریتم و فلوچارت، بهرام غلامی، علیرضا جباریه، انتشارات مؤسسه فرهنگی هنری دیباگران تهران.





Microsoft
Word

Document1 - Microsoft Word

File Home Insert Page Layout References Mailings Review View Add-Ins Nitro Pro

Cut Copy Paste Format Painter Clipboard

Calibri (Body) 11 A A Aa

B I U abc x x² A ab² A

Font Paragraph

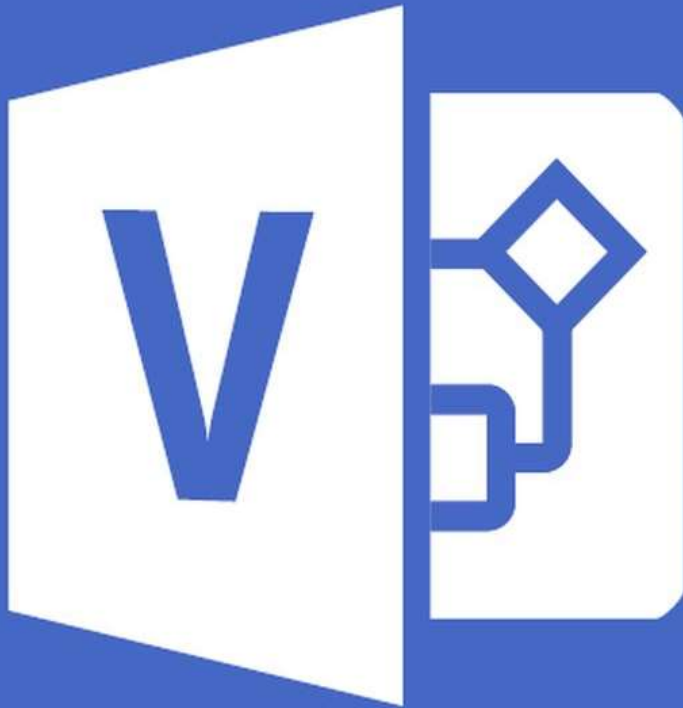
AaBbCcDc AaBbCcDc AaBbCc AaBbCc AaB AaBbCc. AaBbCcDc

¶ Normal ¶ No Spaci... Heading 1 Heading 2 Title Subtitle Subtle Em...

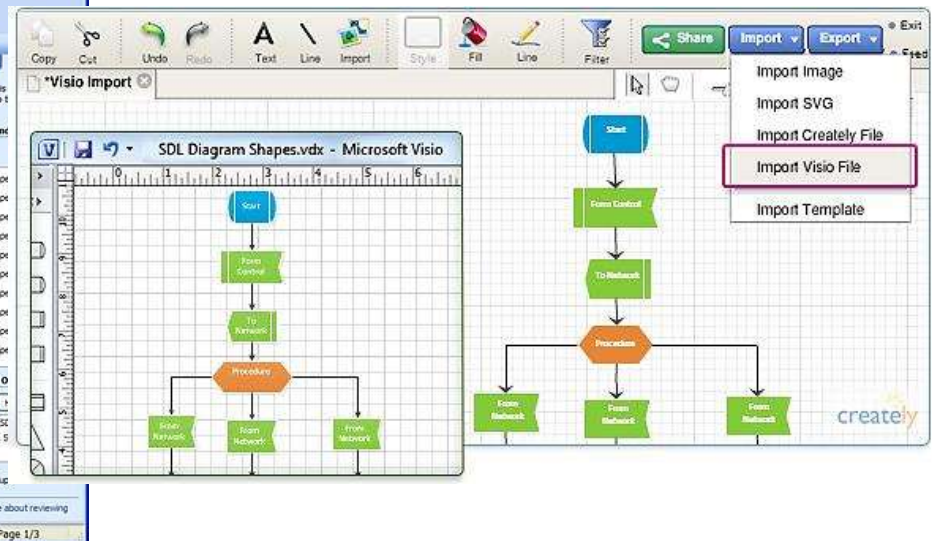
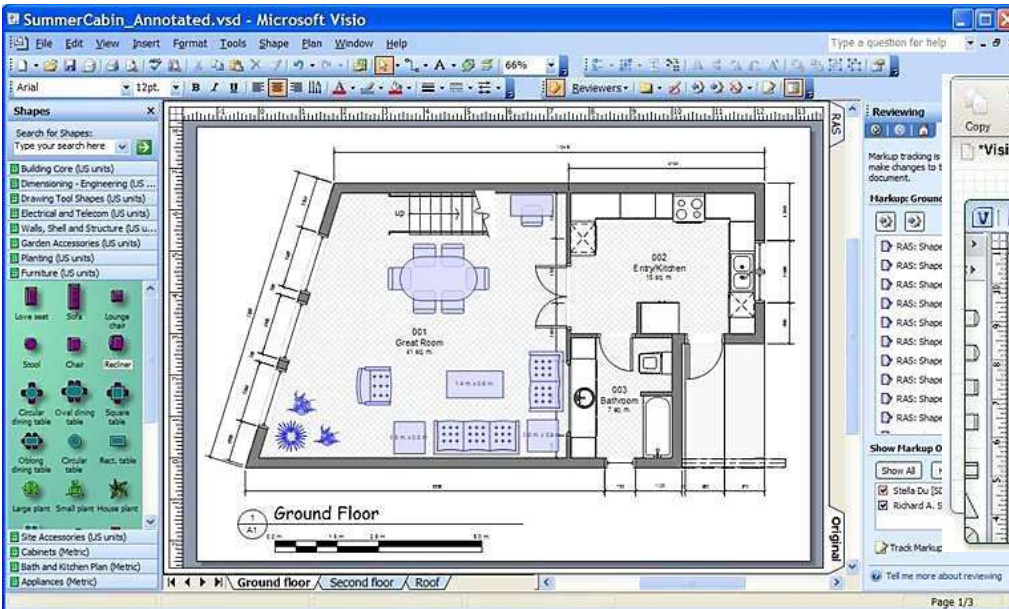
Change Styles

Find Replace Select Editing

72 36 36 72 108 144 180 216 252 288 324 360 396 432 504



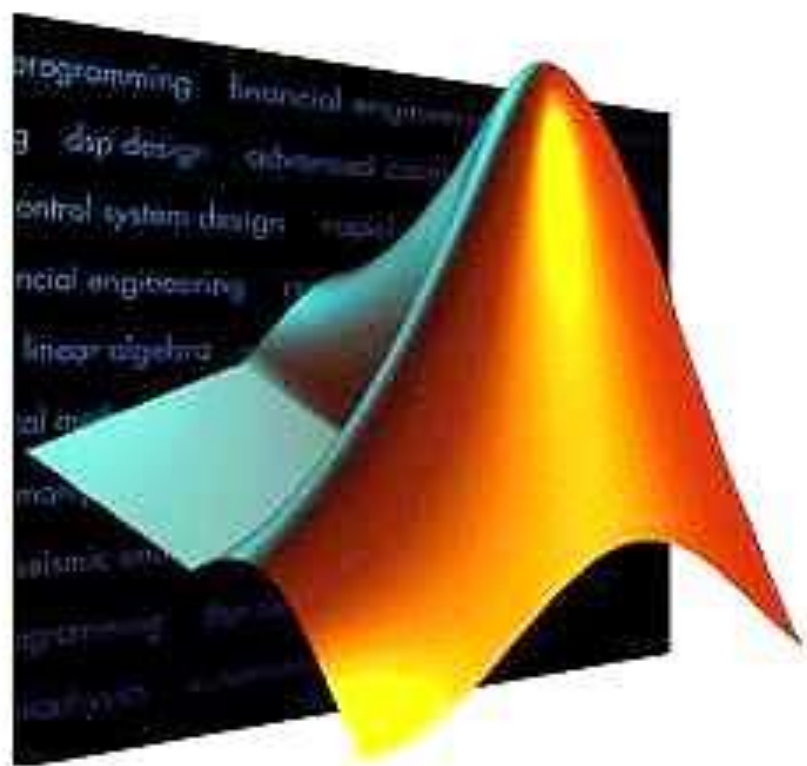
Visio





Microsoft EXcel





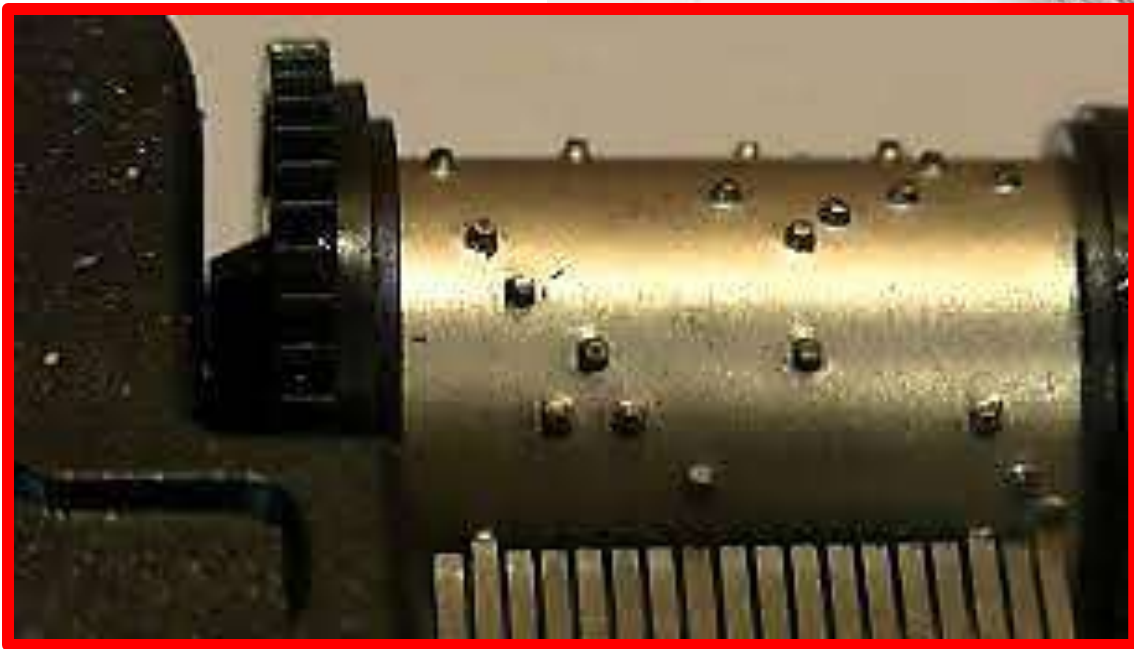
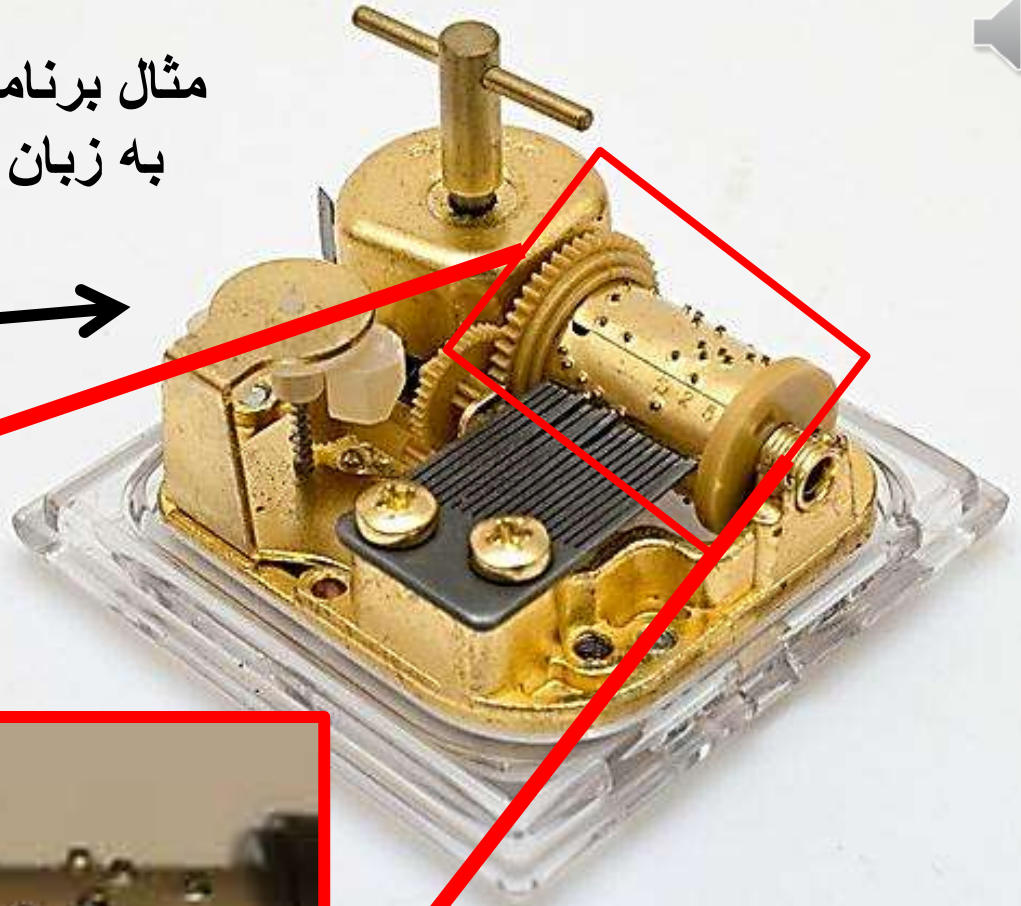
MATLAB[®]

The Language of Technical Computing

The
MATH
WORKS
Inc.



مثال برنامه نویسی
به زبان ماشین



[Return to Page 9](#)