

برنامه نویسی رایانه

مدرس: بهمن رحمتی نژاد

نرم افزار: Matlab

سرفصل ها:

① مقدمه و تاریخچه مختصر رایانه

② اجزای سخت افزار (پردازنده مرکزی، حافظه اصلی، امکانات جانبی)

③ زبان برنامه نویسی و انواع آن (ماشین، اسمبل، سطح بالا)

④ تعریف نرم افزار و انواع آن (سیستم عامل و انواع آن، برنامه ها)

مترجم، برنامه ها، کاربرد)

⑤ مراحل حل مسئله (تعریف مسئله، تحلیل مسئله، تجزیه مسئله)

به مسائل کوچکتر و تعیین ارتباط آنها)

⑥ الگوریتم: تعریف الگوریتم، نمودار دادن راه حل و طراحی

الگوریتم، یا الگوریتم به کمک روند نما، یا الگوریتم به کمک شبه کد

، (نبال کرد الگوریتم، مفهوم الگوریتم

⑦ برنامه و حل مسائل: تعریف برنامه، ساختار کلی برنامه، ساختارهای

(الف) سافت های منطق

(ب) سافت های داده ای

(ج) زیر روال ها

اساسی برنامه سازی

⑧ برنامه

Master

* تعریف نرم افزار و انواع آن :

نرم افزار، مجموعه‌ای از برنامه‌ها و رایانه‌ای، رویه‌ها و مستندات است که انجام کارهای مختلف بر روی سیستم رایانه‌ای را بر عهده دارد. عبارت نرم افزار برای نخستین بار توسط جان تئوکی در سال ۱۹۵۸ مورد استفاده قرار گرفت.

* نرم افزار شامل سه دسته اصلی می‌باشد.

Windows

- ① سیستم عامل : که معمولاً رابط بین کاربر و کامپیوتر می‌باشد مانند Windows
- ② زبانهای برنامه نویسی : که معمولاً افرادی که توانایی پیاده سازی برنامه‌های کامپیوتری را دارند از آنها استفاده می‌نمایند مانند زبان C
- ③ برنامه‌های کاربردی : که توسط برنامه نویسان جهت کاربرد خاص تهیه و در اختیار مشتری قرار می‌گیرد مانند نرم افزار word

* جایگاه نرم افزارها در سیستم رایانه‌ای



* تعریف سیستم عامل: نرم افزار است که مدیریت منابع رایانه را بر عهده گرفته و بهترین راه را برای کار کردن کاربر در اجرا شده و از خدمات آن استفاده کنند. سیستم عامل جزء ضروری ترین نرم افزارها در یک سیستم کامپیوتری است.

* سیر تکاملی سیستم عامل:

- ① Dos ② windows 3.1 ③ windows 95
- ④ windows 98 ⑤ windows 98SE
- ⑥ windows ME ⑦ windows 2000
- ⑧ windows xp ⑨ windows Vista
- ⑩ windows 7 ⑪ windows 8, 10

* نکته: سیستم عامل مکینتاش برای انجام کارهای گرافیکی و سیستم عامل یونیکس و لینوکس برای کاربردهای شبکه این بیشتر مورد استفاده قرار می گیرد.

★ زبان های برنامه نویسی و انواع آن:

زبان های برنامه نویسی ساختارهای زبانی دستورمداری در رایانه هستند که به وسیله آن های توان یک الگوریتم را به وسیله ساختارهای دستوری متفاوت برای اجرا را رایانه توصیف کرد و با این روش امکان نوشتن برنامه جهت تولید نرم افزارهای جدید به وجود می آید. معمولاً هر زبان برنامه نویسی دارای یک محیط نرم افزار برای وارد کردن متن برنامه، اجرا، همگردان و رفع اشکال آن هستند.

★ معمولاً زبان های برنامه نویسی را در پنج نسل تقسیم می کنند.

① نسل اول زبان ماشین است همکار زبان صفر و یک

② نسل دوم زبان های مانند اسمبل و ماسک آن است که قابل فهم تر برای انسان می باشد

③ نسل سوم زبان های مانند کوبول (COBOL) و بی ال

وان (PL1) است که دستورات قابل فهم تر برای انسان و نیاز به کامپیورها

(تقریباً کامیاب: برنامه یا مجموعه‌ای از برنامه‌ها که بهترین است که
متن از زبان برنامه نویسی سطح بالا (زبان مبدا) را به زبان سطح پایین
(زبان مقصد) ~~تبدیل~~ تبدیل می‌کند)

۴) نسل چهارم زبان‌های اوریکل (Oracle) و فاکس پرو (Fox Pro)
واس کیوال‌ها (SQL) است و این نسل چیزی نزدیک
به محاوره‌ها انسان است.

۵) نسل پنجم زبان‌ها مانند Prolog، OPS5، و غیره ال بیسیک
تمرکز بر حل مسئله و استفاده از الگوریتم‌ها نوشته شده، نوکله‌ها
برنامه نویسی.

★ زبان‌ها برنامه نویسی سطح پایین و سطح بالا و لزوم استفاده از کامپایلر
★ زبان برنامه نویسی سطح پایین و این زبان مستقیم باید دانشنده‌ی سیستم
سروکار دارد. و با استفاده از آن‌ها می‌توان دستورات پایه‌ای
برنامه نویسی را اجرا کرد. فصحیه دستورات نوشته شده در
این زبان‌ها کار دشوار است. به طور مثال، یکم از این زبان‌ها

سطح پایین Machine language یا زبان ماشین نام دارد.
 نکته جالب آن که از دستورات انگلیسی در آن استفاده شده باشد،
 از اعداد صفر و یک برای نوشتن دستورات در آن استفاده می‌شود.
 زبان اسمبل هم یک نمونه‌ی دیگر از زبان‌های سطح پایین است.
 * زبان‌های برنامه‌نویسی سطح بالا:

در این برنامه‌ها از دستورات همچون کلمات انگلیسی که برای انسان
 قابل فهم‌تر هستند، استفاده می‌شود. مانند زبان‌های پایتون، جاوا،
~~اسکرپت~~ اسکریپت، پی‌ایچ‌پی، رپ، فورتران، پورپیک، پایسکال
 * لزوم استفاده از کامپایلر:

زبانی که برنامه‌نویسی دستورات مد نظرش را در یکی از زبان‌های
 برنامه‌نویسی سطح بالا مثل پایسکال، پورپیک می‌نویسد، نیاز به نرم
 افزار است که آن کدها را به زبان ماشین یا همان صفر و یک ترجمه
 کرده و در اختیار سیستم قرار دهد. چرا که سیستم‌ها فقط و فقط به
 زبان ماشین یا همان صفر و یک را متوجه می‌شوند.

مراحل حل مسئله نوک کا مہیترہ

جدا کر دے

① تعریف صورت مسئلہ بہ صورت جامع و دقیق (انتزاع)

② تعیین راه حل قدم بہ قدم (الگوریتم) ^{درآوردن جزئی از یک کل} براں حل مسئلہ

③ تبدیل راه حل قدم بہ قدم با الگوریتم بہ دادہ ساختار (ساختار دادہ)

④ بررک درستی (تحلیل)

⑤ نوشتن یک برنامه با استفادہ از یک زبان / برنامه نویسی

⑥ عیب یابی برنامه و اطمینان از صحت کارکرد

★ تفاوت ساختار دادہ و الگوریتم

ساختار دادہ ہماروش ذخیرہ دادہ ہا در رایانہ باهدف دہترکی
آسان تر و ہمینہ تر است در حالیکہ الگوریتم روشی بہ منظور حل
مسئلہ بہ وسیلہ کا پیتر است.

تعریف الگوریتم: مجموعه‌ای از دستورالعمل‌ها است، که به ترتیب خاص اجرا می‌شوند و مسئله‌ای را حل می‌کنند به عبارت دیگر یک الگوریتم روشی است که برای حل مسئله است. این روش برای نخستین بار توسط خوارزمی مورد توجه قرار گرفت. شیوه محاسبه معدل در مدرسه، یکی از نمونه‌های الگوریتم است.

★ خصوصیات یک الگوریتم:

- ① ورودی: یک الگوریتم یک یا چندین پارامتر را به عنوان ورودی دارد.
- ② خروجی: الگوریتم باید حداقل یک گیت به عنوان خروجی (نتیجه عملیات) تولید کند.

- ③ قطعیت: دستورهای الگوریتم باید با زبان دقیق و بی‌معنی انجام پذیر باشد. هر دستور العمل نیز باید انجام پذیر باشد (مثلاً حاصل تقسیم پنج بر صفر را محاسبه کنید مجاز نمی‌باشد چون این عملیات در ریاضیات تعریف نشده است)

④ محدودیت: الگوریتم باید دارای شروع و پایان مشخص باشد، به نحوی که اگر دستورهای آن را دنبال کنیم، برای تمام حالات، الگوریتم پس از طی مراحل، خاتمه یابد. به علاوه، زمان برای خاتمه الگوریتم هم باید به گونه ای محلول و کوتاه باشد.

* روش های بیان الگوریتم:

① بیان الگوریتم ~~به صورت گرافیکی~~ (شبه کد)

② بیان ریاضی الگوریتم (۳) بیان الگوریتم توسط اشکال (زئوگرام)

* تعریف شبه کد (Pseudocode) سودود

روشی برای توصیف ~~عملیات~~ عملکرد و بیان گام های یک الگوریتم یا برنامه رایانه ای می باشد. در این روش از ترکیب برخی ویژگی های ساختار زبان های برنامه نویسی در کنار زبان طبیعی استفاده می شود و تلاش می شود تمام آنچه برای عملیاتی نمودن، مورد نیاز است به شکل خلاصه و قابل فهم بیان شود. در عین حال از ~~سودود~~ ^{notebook} ARPACH

جزئیاتی نظیر تعریف متغیرها در روش شبیه‌سازی خود داریم می‌شود
این روش در زبان ماس برنامه نویسی پیمیک، C، پاسکال
و ++C استفاده می‌شود.

* بیایم ریاضی الگوریتم: در این روش از توابع و روابط ریاضی جهت
تجزیه و تحلیل الگوریتم استفاده می‌کنیم.
* فلوچارت:

فلوچارت مجموعه‌ای از علائم واسطه‌های خاص است که الگوریتم
را به صورت نمادهای تصویری نمایش می‌دهد.
* علائم فلوچارت

① علامت شروع و پایان: در فلوچارت از بیضی برای شروع و بیضی
الگوریتم استفاده می‌شود (شروع) (پایان)
② علامت جابجایی و انتساب

در فلوچارت برای انجام عمل جابجایی و یا انتساب و یا عمل

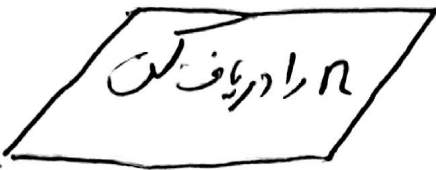
محاسباتی از مستطیل استفاده می شود.

$i \rightarrow 1$

(۳) علامت ورودی:

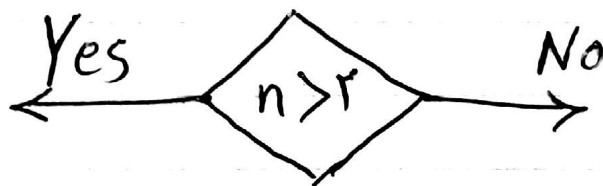
در فلوچارت از علامت متوازی الاضلاع برای گرفتن ورودی ها

استفاده می شود.



در مستطیل

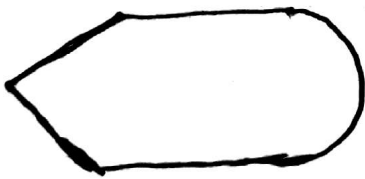
(۴) علامت شرط: در فلوچارت از علامت لوزی برای شرط استفاده



می شود.

(۵) علامت چاپ:

در فلوچارت از علائم زیر برای چاپ استفاده می شود.



چاپ در مستطیل

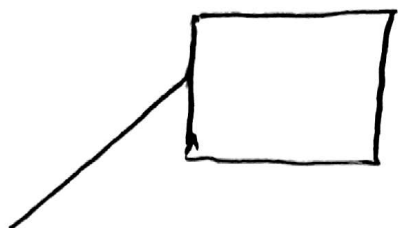


چاپ در کاغذ

(۶) علامت فلش: از فلش های جهت دار به منظور ارتباط

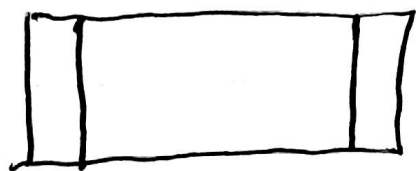
بین علائم استفاده می شود.

⑦ حاشیه نویسی: حاشیه نویسی برار نمایش نظرات یا نکات قابل توجه در کنار فلوجارت استفاده می شود. و از علامت زیر استفاده می شود.



⑧ دستورات از پیش تعیین شده:

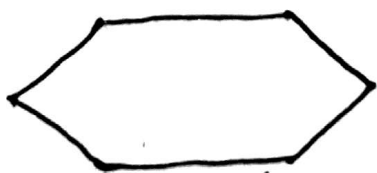
یک مستطیل با دورش موازی و عمود برار نمایش قدم ها را مشخص می کند. و آینه استفاده می شود که ممکن است در فلوجارت جداگانه این



تعریف شده باشند

⑨ آماده سازی:

یک شش ضلعی، می تواند برار مقدار در اول استفاده شود. همچنین این نماد ممکن است به جای نماد شرط و تصمیم گیری



استفاده شود.

⑩ اتصال دهنده: معمولاً به شکل دایره نمایش داده می شود و نشان می دهد که جایی که چند گانه کنترل، در یک جایگاه قرار

خارج می شود

۱۱) اتصال دهنده: مانند شماره ۱۰۱ عمل می کند با این تفاوت که می تواند
برای جان گذاری یک اتصال دهنده به صفت دیگر استفاده شود.
مثال: الگوریتم برای پیدا کردن دو عدد ۱۰ و ۲۰ را بنویسید

الف) به روش شبه کد

ب) روش خوارچارت

الف) شبه کد

۱) شروع

۲) دو عدد ۱۰ و ۲۰ را بگیر

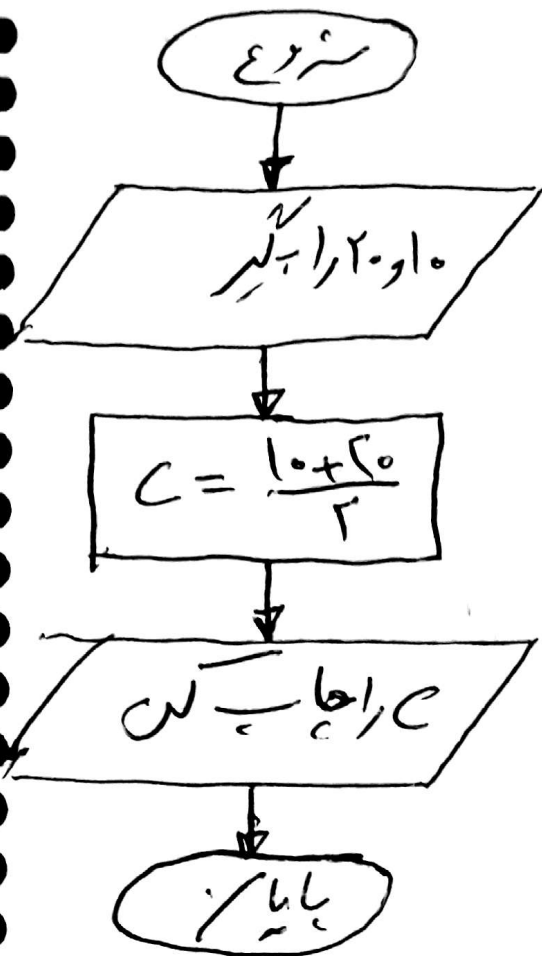
۳) $\frac{10+20}{2}$ را محاسبه کن و در مکان (متغیری) به نام C قرار بده

۴) C را چاپ کن

۵) پایان

نکته: شبه کد: به زبان الگوریتم با استفاده از کلمات، عبارات و علائم ریاضی

(ب) فلوچارت:



★ اگر در تمام فلوچارتی را بنویسید که عددی را دریافت کند و آن عدد زوج بود
کدام زوج و آن فرد بود کلمه فرد را در خروجی چاپ کند

① شروع کن ② عدد A را دریافت کن

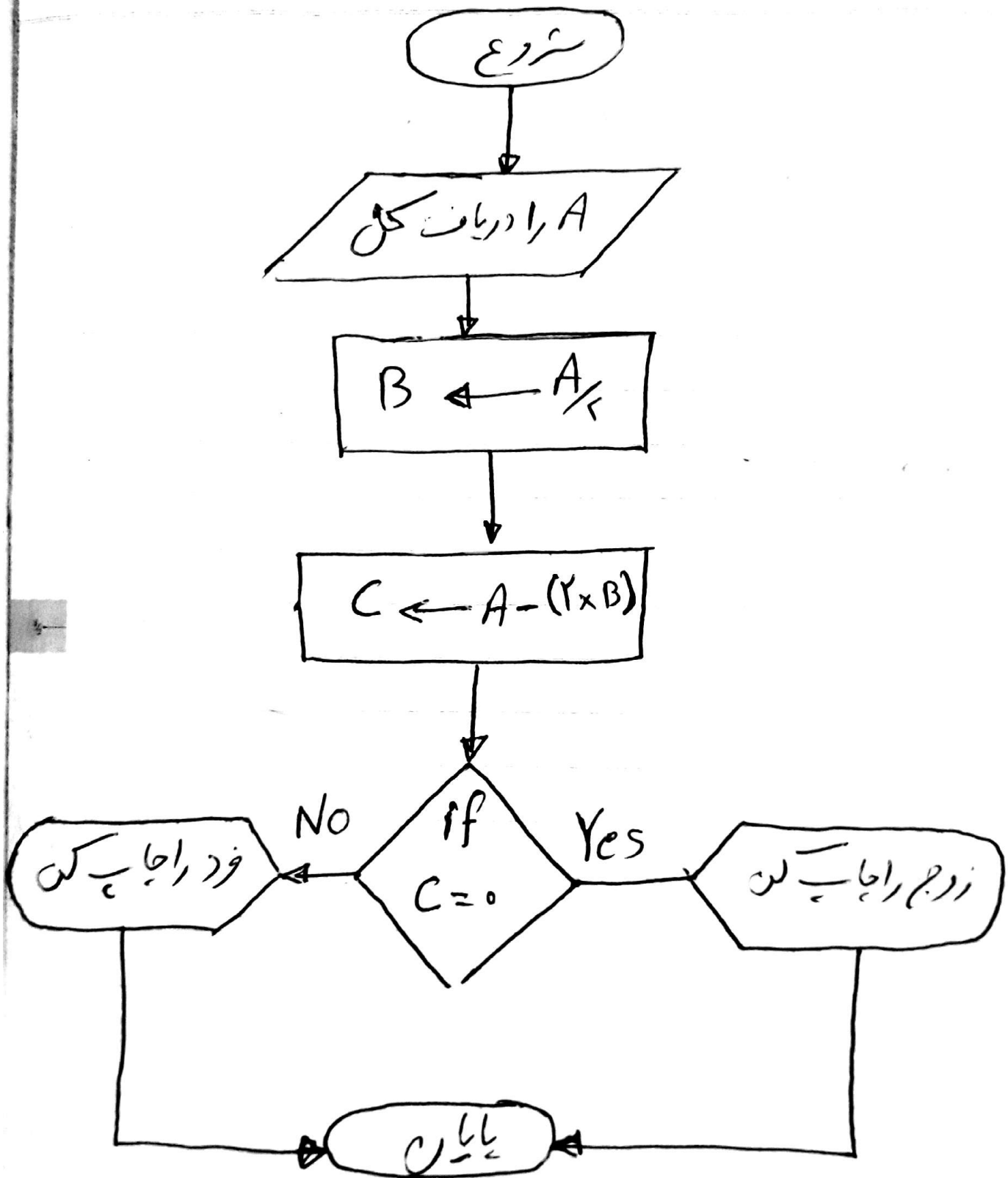
③ $B \leftarrow \frac{A}{2}$ ④ $C \leftarrow A - (2 \times B)$

⑤ اگر C برابر با صفر بود آنگاه در خروجی زوج را چاپ کن در غیر این صورت فرد

$$\frac{12}{2} = 6 = B$$

$$C = 12 - (2 \times 6) = 0$$

⑥ پایان



★ مثال: نعلو چارتن رسم نمائید که جواب معادله درجه اول یک مجهولی

$$ax + b = c \quad (a \neq 0) \text{ را محاسبه و درون صفحه کاغذ چاپ نماید.}$$

الگوریتم
(۱) شروع

(۲) a را بگیر

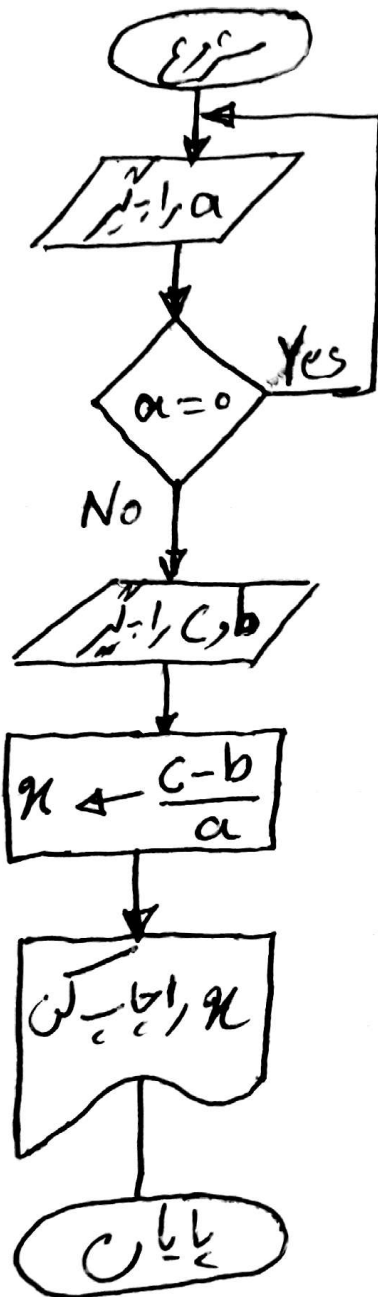
(۳) اگر $a = 0$ پس برو به ۲

(۴) b و c را بگیر

$$x \leftarrow \frac{c-b}{a} \quad (۵)$$

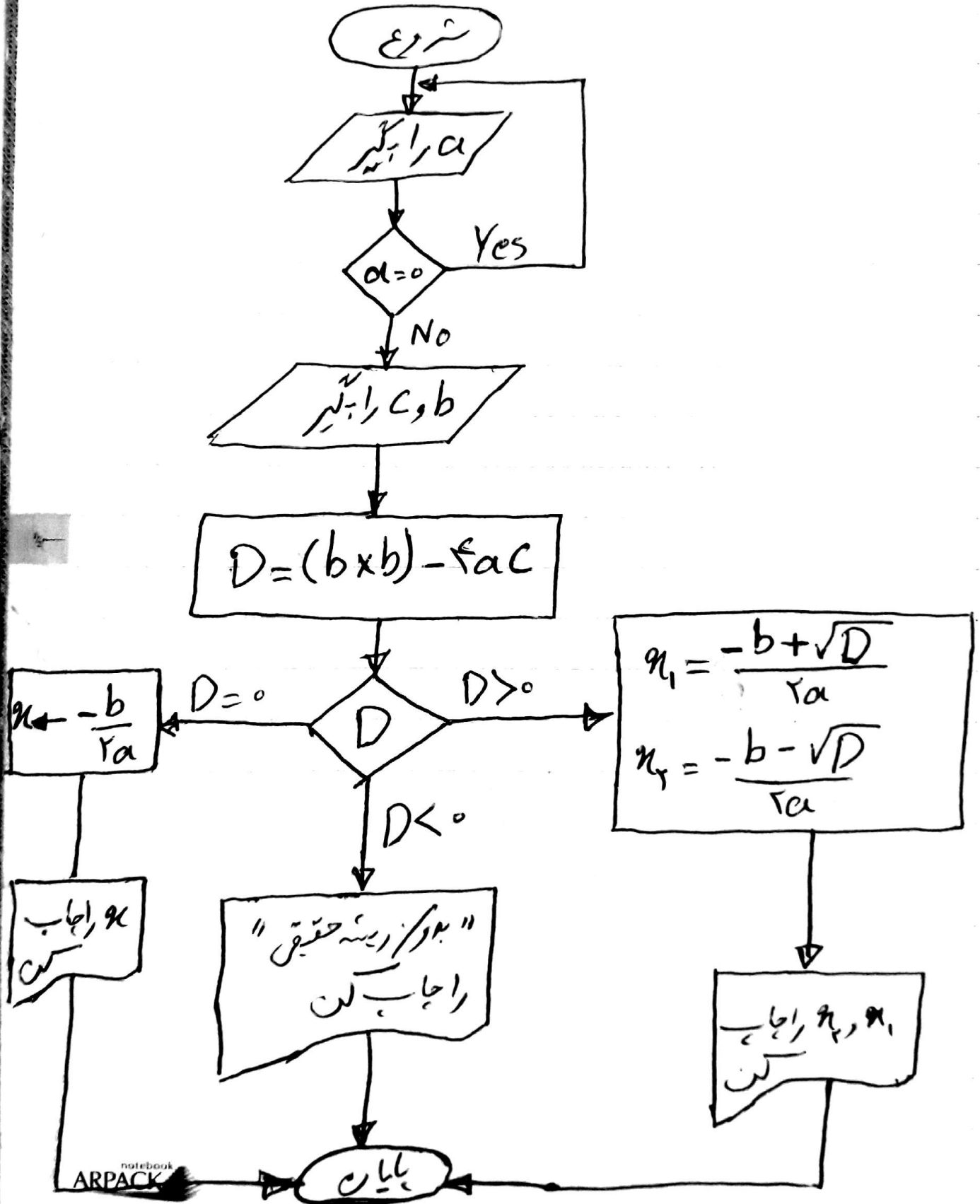
(۶) x را چاپ کن

(۷) پایان



۱۷

* الگوریتم و فلوجارت برای حل یک معادله درجه یک را بنویسید.



الگوریتم تعیین جواب معادله درجه ۲

① ضرایب a ، b و c را از ورودی بخوان

② دلتا را با فرمول $D = b^2 - 4ac$ محاسبه کن

③ شرط: اگر $D < 0$ باشد نمایش بده: "معادله ریسه حقیقی ندارد" بر روی پایانه

④ شرط: اگر $D = 0$ باشد "معادله ریسه مضاعف دارد و مقدار آن عبارتست

است از $x = -\frac{b}{2a}$ بر روی پایانه

⑤ شرط: اگر $D > 0$ باشد نمایش بده "معادله دارای دو ریسه حقیقیست

که مقدار هر یک عبارتند از:

$$x_1 = (-b + D^{1/2}) / 2a$$

$$x_2 = (-b - D^{1/2}) / 2a$$

بر روی پایانه

⑥ پایانه

معرفی نرم افزار MATLAB

یکی از زبان‌های سطح بالا و پهنای محیط جذاب می‌باشد که در ابتدا بر اساس زبان برنامه نویسی C توسعه داده شد. Matlab از ترکیب دو واژه MATrix (ماتریس) و LABoratory (آزمایشگاه) ایجاد شده است. دلیل استفاده از نام آزمایشگاه ماتریس برای این نرم افزار این است که تمام محاسبات و شبیه سازها و داده‌های خود را از طریق ماتریس انجام می‌دهد. خصوصیت بارز این نرم افزار دارای بودن شبیه ساز در اکثر علوم فنی و مهندسی می‌باشد.

★ برخی از کاربردهای متلب

① پردازش سیگنال و ارتباطات

② پردازش تصویر و ویدئو

③ سیستم‌های کنترل ④ تست و اندازه‌گیری

⑤ مهندسی مالی ⑥ محاسبات زیستی

* برخی از محاسبات ریاضی که در مطلب بیشتر استفاده می شود

① کار با ماتریس و آرایه ها

② رسم نمودارها در دو بعد و سه بعد و ترانزیس

③ جبر خطی ④ معادلات جبر ⑤ توابع غیر خطی

⑥ آمار ⑦ آنالیز داده ⑧ حساب دیفرانسیل و انتگرال

و معادلات دیفرانسیل ⑨ محاسبات عددی

~~در~~ در اصل نصب نرم افزار MATLAB 2008

① ابتدا اتصال به اینترنت را قطع کنید

② در پشته برنامه Setup را اجرا کنید

③ در اولین پنجره گزینه

• Install manually without using the Internet

را انتخاب کنید

④ در پنجره بعدی در جواب

Do you accept the terms of the license agreement

جواب Yes، انتخاب کنید

⑤ سپس با انتخاب گزینه

• I have the file Installation key for my license

کد زیر را وارد کنید ۵۲۶۲۶ - ۱۱۱۱۱ - ۱۱۱۱۱ - ۱۱۱۱۱

⑥ در پنجره جدید Typical را انتخاب کنید

⑦ سپس مسیر نصب برنامه را مشخص کنید. فایل مورد نیاز

برای اجرا برنامه ۴۳۶.۰۴ مپ است.

⑧ با انتخاب مسیر نصب و Next عبور کنید پنجره تأیید می آید

که باید Yes را انتخاب کنید

⑨ در پنجره باز Install را انتخاب کنید تا مراحل نصب

شروع شود

⑩ بعد از اتمام مراحل نصب پنجره باز Next کنید

⑪ در پنجره جدید ☒ Active MATLAB را انتخاب کنید

Next را بزنید

⑫ در پنجره جدید

• Active manually without the Internet

را انتخاب کنید

(۱۳) در پنجره باز شده گزینه
Enter the path to the license file

را انتخاب کرده و باز در Browsers از پوشه Crack فایل
license.dat را انتخاب کنید

(۱۴) پنجره را Next عبور و finish را انتخاب کنید

★ نکته مهم: با توجه به اینکه نرم افزار مبتنی بر جین اجرا تاریخ سیستم
را چک می کند. قبل از اجرا تاریخ سیستم را به روز ۲۰۰۸ تنظیم
کنید. تا در جین اجرا نرم افزار به مشکل برخوردید.

★ معرفی محیط اصلی نرم افزار MATLAB

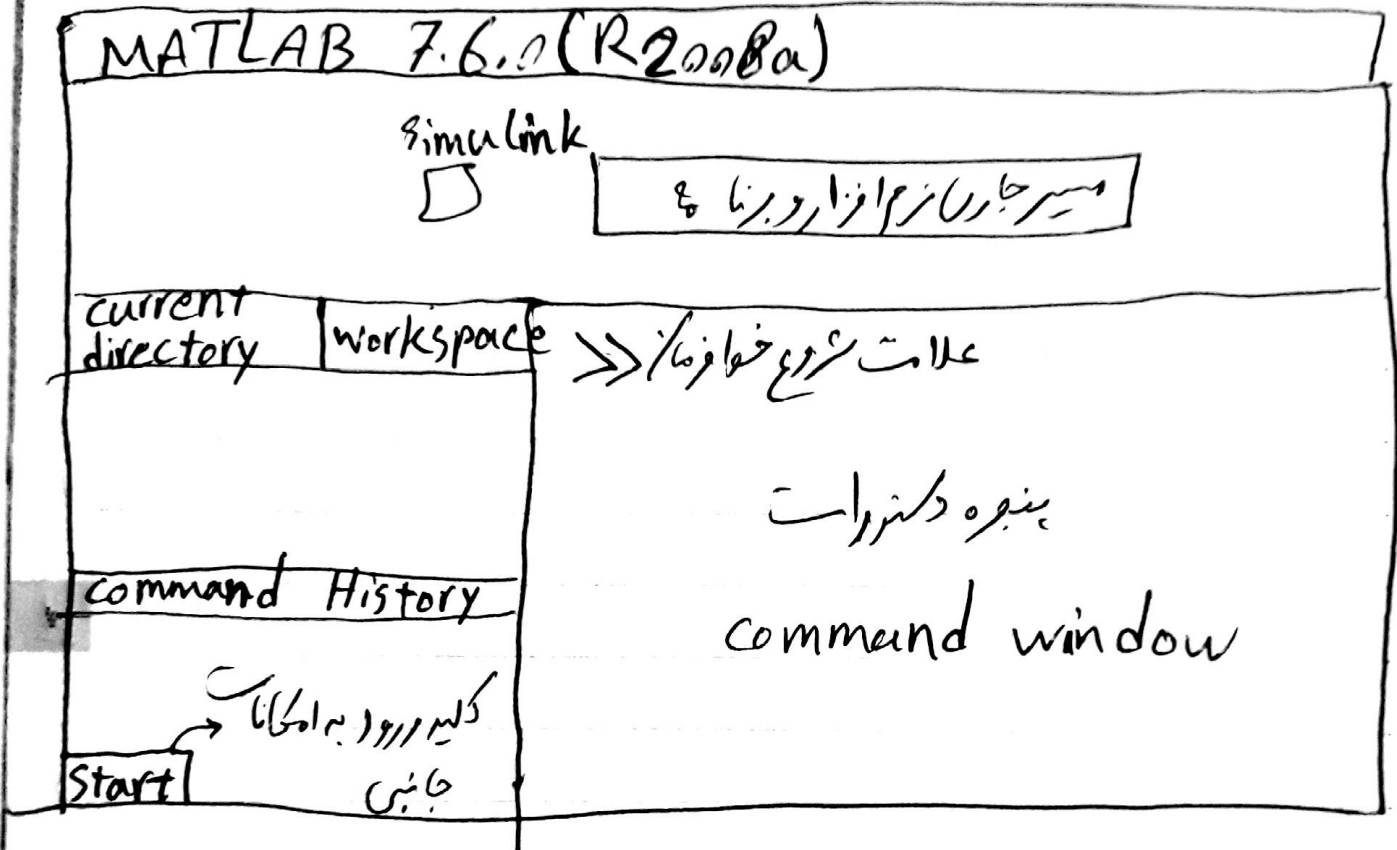
پنجره Current Directory: محتوی نام فایل های مسیر جاری و

نشانی درمنه محل اجرا برنامه ها M-file است.

پنجره Command History: شامل لیست تمام دستورات

می باشد که تاکنون پس از اجرا نرم افزار از آنجا استفاده کرده اید

نکته: برای پاک کردن این لیست می‌توانید در این پنجره کلیک راست کرده و گزینه `clear Entire History` را انتخاب نمود.



* پنجره `workspace`: در صفحه کار نرم افزار، می‌توان نام متغیرهای استفاده شده و مقدار درون متغیر و کلاس متغیر و حتی کمترین و بیشترین مقدار آن که در درون متغیر قرار گرفته است را مشاهده کرد. پنجره `command window`: برای تایپ دستورات و مشاهده

* M-file: برای نوشتن برنامه‌ها می‌توان از این محیط استفاده نمود. برای ورود به این محیط باید Command window فعال باشد پس از سرزیر M-file را باز کنید.

① File \ New \ M-file

② Ctrl + N

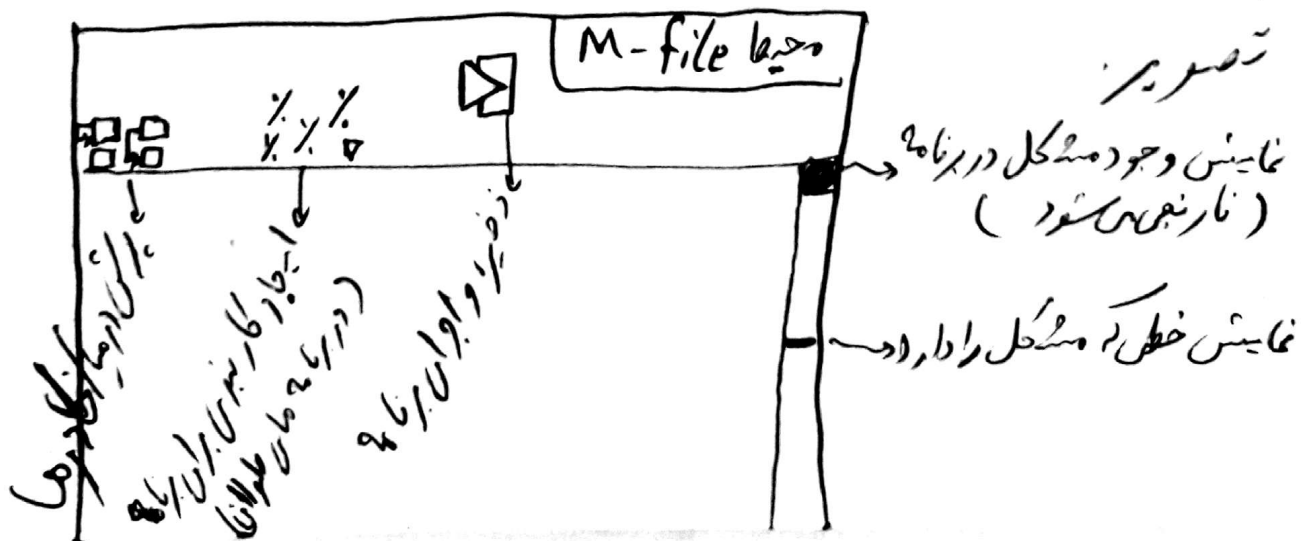
از این محیط می‌توان به دو صورت استفاده کرد.

④ برنامه نویسی Function (تتابع نویسی): زمانی که بخواهیم

در محیط متلب تابعی را معرفی کنیم (تابعی که در کتابخانه متلب نباشد) و

در برنامه ~~معمول~~ به صورت مکرر از آن استفاده کنیم

⑤ برنامه‌ها را اجرای (Script): مانند نوشتن برنامه‌ها را از



نکته: در صورتی که می‌خواهید خطی از برنامه را به توضیحات اختصاص دهید و یا در زمان اجرای برنامه نیاز دارید خطی از برنامه را غیر فعال نمایید می‌توانید از علامت $\backslash$ در ابتدای خط مورد نظر استفاده کنید.

* شرایط تعریف متغیر در متلب:

① متلب به حروف بزرگ و کوچک حساس است. پس متغیر a کوچک با متغیر A بزرگ تفاوت دارد.

② حرف اول یک متغیر نباید از عدد استفاده شود.

③ نام متغیر حداکثر ۳۱ تا ۳۶ کاراکتر می‌تواند باشد.

④ نام متغیرها را نباید از کلمات تعریف شده در متلب باشد. البته دیکشنری

متلب با حروف کوچک نوشته می‌شود و شما می‌توانید برای نامگذاری متغیر

خود از این نامها با حروف بزرگ استفاده کنید.

⑤ بین حروف یک کلمه نباید فاصله وجود داشته باشد.

⑥ از underline در نامگذاری متغیرها می‌توان استفاده کرد.

اما از کاراکترهای علامت و دیکشنری نمی‌توان استفاده کرد.

* ایجاد ماتریس m سطر و n ستون در متلب :

اعضای یک ماتریس m در n را باید در یک درآلت [] معرفی کرد.

هر سطر را با علامت ز از یکدیگر جدا کنید و برای جدا سازی کتورهاها

(اعضای هر سطر) بی تران از فاصله یا علامت = استفاده نمود.

مثال : ماتریس $\begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$ را ایجاد و در متغیر a قرار دهیم

$$\gg a = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$$

* دستیابی به آرایه ان مختص از یک ماتریس :

$$A = a(x, y)$$

از ماتریس a آرایه سطر x و ستون y را جدا کرده و در متغیر A قرار می دهیم.

$$A = a(:, y)$$

از ماتریس a سطر x را جدا کرده و در متغیر C قرار می دهیم.

$$C = a(x, :)$$

از ماتریس a کتور y را جدا کرده و در ماتریس C قرار می دهیم.

۲۷

میل

$\rightarrow C = a(2, 3)$

$C = V$

$\rightarrow C = a(2, :)$

$C = \wedge \quad a \quad 10$

$\rightarrow C = a(:, 1)$

$C = 2$
 $\wedge$

$\rightarrow C = a(1:2, :)$

$C = \begin{matrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{matrix}$

$\rightarrow C = a(1:3, :)$

$C = \begin{matrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ \wedge & a & 10 \end{matrix}$

notebook
ARPACK

★ ایجاد ماتریس با استفاده از توابع:

★ ماتریس واحد (ماتریس (تعداد سطر / تعداد ستون و تعداد قطر
قطر)

$$\gg a = \text{eye}(3, 3)$$

$$a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

★ ماتریس یک:

$a = \text{ones}$ (تعداد سطر / تعداد ستون)

$$\gg a = \text{ones}(3, 4)$$

$$a = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$a = \text{zeros}$ (تعداد سطر / تعداد ستون)

★ ماتریس صفر:

$$a = \text{zeros}(3, 4)$$

$$a = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

★ ماتریس راندم: اعداد اشتقاق بین صفرتا ۱ خواص در بردار کاربرد این
ماتریس در وارد کردن نویز به سیستم است.

» $a = \text{rand}(\text{تعداد سطر}, \text{تعداد ستون})$

$a = \text{rand}(3, 4)$

جواب به صورت اعداد مثبت اتفاق نامی رقم اعشاری باشد.

$a = \text{randn}(\text{تعداد سطر}, \text{تعداد ستون})$

جواب به صورت اعداد مثبت و منفی اتفاق نامی رقم اعشاری باشد.

* ماتریس مربعی جا دوی: ماتریس مربعی و اتفاق که نتیجه جمع سطرها

یا ستونها برابر باشد. $a = \text{magic}(\text{تعداد سطر و ستون})$

» $a = \text{magic}(3)$

$a =$	۸	۱	۶
	۳	۵	۷
	۴	۹	۲

* دستور `whos`; این دستور لیست تمامی متغیرهای موجود

استعداد، رایج همراه ابعاد و مقدار حافظه اشغال شده در برنامه و حقیقی یا

whos

Name

size

Byte

class

Attributes

ARPACK

۴x۴

۱۲۸

double

—

c

۳x۳

۷۲

~

غایب من آمد.

۲۵

★ دستور Size: این دستور اندازه ماتریس را به صورت دو عدد که اولی

تعداد سطرها و دومی تعداد ستونها را نشان می دهد (mm)

$$e = \text{size}(a) \quad \leftarrow$$

$$c =$$

★ دستور Length: این دستور تعداد آرایه ها را در یک سطر

ماتریس حساب کرده و آن را در متغیر گفته شده قرار می دهد.

فرض کنید $a = \begin{bmatrix} 2 & 6 & 5 & 4 & 8 \\ 1 & 2 & 5 & 7 & 9 \end{bmatrix}$

$$e = \text{Length}(a) \quad \leftarrow$$

$$c = 5$$

★ دستور disp: این دستور بیشتر در M-file استفاده می شود

و برای نمایش ماتریس یا آرایه بکار می رود

$$\gg \text{disp}(a) \quad \leftarrow$$

ماتریس a نمایش داده می شود.

نکته: با این دستور متراکم در نیاز خود را می توانید با قرار دادن در بین

علامت $\gg$ به نمایش درآورد
 $\gg \text{disp('Hello')}$

Hello

* * clear : پاک کردن یک متغیر : از این دستور برای پاک کردن یک

یک متغیر که قبلاً تعریف شده است استفاده می شود.

$\gg \text{clear a}$

clear all

نکته : اگر clear به تنهایی نوشته شود تمام متغیرها را پاک می کند

* * دستور CLC : از این دستور برای پاک کردن پنجره Command window

استفاده می شود
 * عملگرها ریاضی (+, -, *, /, ^, ...)

عملگرها در مثال هم می توانند به کل آرایه اعمال شود و هم می توانند به تک تک آرایه ها اعمال شود و زمانی که عملگرها به تک تک آرایه ها اعمال می شود از علامت { استفاده می شود. یک مثال توضیح می دهیم.

مثال : دو ماتریس زیر را در نظر بگیرید.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$B = \begin{bmatrix} 2 & 5 \\ 2 & 2 \end{bmatrix}$$

تک تک آرایه ها به هم ضرب شده اند

دو ماتریس به هم ضرب شده اند

$$\gg A * B \quad \text{ans} \quad \begin{matrix} 4 & 4 \\ 14 & 14 \end{matrix}$$

$$\left. \begin{matrix} A \circ B \\ \text{ans} \end{matrix} \right\} \begin{matrix} 2 & 4 \\ 4 & 8 \end{matrix}$$

★ دستور sum: این تابع که حاصل آن یک ماتریس سطری است، آرایه‌های موجود در دستور/ماتریس را جمع کرده و حاصل را در متغیر

تعریف شده قرار می‌دهد (مدرسه)

$$a = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$
 فرقی نشود

» $c = \text{sum}(a)$ ←

$c = 3 \quad 5 \quad 7 \quad 9 \quad 11$

★ دستور plot: از این دستور برای ترسیم ماتریس‌های خطی، بردارها و منحنی‌ها

توابع استفاده می‌شود.
 » $\text{plot}(a)$ ←

★ دستور mesh: از این دستور برای ترسیم ماتریس به صورت

سه بعدی استفاده می‌شود.
 » $\text{mesh}(a)$ ←

★ دستور max: این تابع که حاصل آن یک ماتریس سطری می‌باشد،

آرایه‌های موجود در دستور/ماتریس را مقایسه و ماکزیمم مقدار هر سطر/

را در متغیر تعریف شده قرار می‌دهد.

۲۲۹

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 10 & 5 & 4 \\ 2 & 3 & 0 \\ 9 & 8 & 7 \end{bmatrix}$$

$\gg \max(a) \leftarrow$

ans =

10 8 7

* تابع min: برعکس دستور قبل بوده و کمترین مقدار را انتخاب می‌کند

$\gg \min(a) \leftarrow$

ans =

1 2 3

* نکته مهم: اگر max و min یک هم‌رود نظر باشد می‌تواند ماتریس را ۹۰ درجه چرخاند پس max و min را مبادی کرد

* میانگین/mean: میانگین هر سطر را مبادی می‌کند

$\gg \text{mean}(a) \leftarrow$

ans =

5,5 4,5 5,25

* چرخش ماتریس ۹۰: از این دستور برای چرخش ماتریس در جهت

خلاف عقربه‌های ساعت استفاده می‌شود

$c = \text{rot90}(a) \leftarrow$ چرخش ماتریس به اندازه ۹۰ درجه

$$c = \begin{bmatrix} 2 & 4 & 5 & 7 \\ 5 & 3 & 8 \\ 1 & 10 & 2 & 9 \end{bmatrix}$$

* در این حالت جایی سطر و ستون عوض می‌شود

۳۴

دستور diag: قطریه مایه‌های را نشان می‌دهد.

>> diag(a)

ans =
1
5
5

* در میان مایه‌های: det: در مایه‌های یک‌بار از این دستور برای پیدا کردن در میان استفاده می‌شود.

>> det(a) ans = -49
 $a = \begin{bmatrix} 1 & 1 & 4 \\ 2 & 5 & 7 \\ 8 & 9 & 5 \end{bmatrix}$ فرض کنید

* دستورات حلقه و شرطی:

این نوع دستورات برای به وجود آوردن یک شرط برای اجرای یک عملیات و یا برای ساخت حلقه برای انجام کارهای تکراری و خاص برنام‌نویس می‌باشند.

* دستور حلقه for:

عدد پایانی: عدد شروع = نام متغیر for
عملیات مورد نظر
end

باین روش بران یک متغیر مقدار شروع و خاتمه را در نظر می گیرند
 عملیات در هر حلقه که در میان for و end قرار دارد به تعداد متداری
 که بران متغیر در نظر گرفته اید، تکرار می شود. بهر بار پس از end به اول
 حلقه، به متغیر یک اضافه می شود. end بایا حلقه است و در زمانی که
 مقدار متغیر به مقدار بایا در نظر گرفته شده رسیده این حلقه بایا به
 وزنم افزای به خط بعدی و اجرای ادامه برنامه می پردازد. بهر است این دستور
 در M-file نوشته شود.

نکته: بران خروجی ناکامی از حلقه از دستور break استفاده نمایند.
 نکته: بعد از نوشتن M-file فایل را در مسیر جاری Current Directory

ذخیره کنید.
 مثال: برنامه بنویسید که اعداد از یک تا ۵ را با هم دیکر جمع کند و در متغیر B

```
clc
A = 0
for n = 1:5
    A = A + n
end
B = A
```

قرار دهد.
 بعد از نوشتن این برنامه در M-file و ذخیره آن در
 مسیر جاری آکون Run را بزنید.

$$B = 15$$

$$1 + 2 + 3 + 4 + 5 = 15 \quad \text{جواب به صورت دستی}$$

* نکته: شما می توانید استپ ها را تغییر و راحم عوض کنید.

```
clc
A = 0
for n = 1:2:15
    A = A + n
end
B = A
```

↑ استپ ها
↑ شروع
↑ ۱۵

$$B = 25 \quad \text{جواب}$$

$$1 + 3 + 5 + 7 + 9 = 25 \quad \text{جواب به صورت دستی}$$

* دستور حلقه شرطی while

در این روش تا زمانی که شرط (شرط) while

برقرار باشد، حلقه تکرار می شود. عملیات مورد نظر

end

* مثال: در این برنامه تا زمانی که A به عدد بزرگتر از ۱۰ نرسیده

باشد عملیات تکرار می شود.

clc

A=0

while (A <= 10)

A = A + 1

end

B = A

نکته: اگر و گذاشته نشود یا هر بار تکرار جواب چاپ شده و در صورتی که و گذاشته نشود نتایج چاپ نخواهد شد. و در آخر که منتهی به B=A جواب آخر چاپ می شود.

B = 11 جواب

★ دستور pause (ایجاد تأخیر): توقف این دستور به مقدار دلخواه در

برنامه تأخیر می توان ایجاد نمود مقدار تأخیر بر حسب ثانیه وارد می شود.

★ مثال: برنامه ای بنویسید که مثال قبل را با ۱۰ ثانیه اجرا کند.

clc

A=0

while (A <= 10)

pause(10)

A = A + 1;

end

B = A

★ دستور شرطی `if` :
 در این روش زمانی که شرط برقرار باشد، عملیات مورد نظر انجام می شود
 در غیر این صورت توابع بعد از `end` ادامه می یابند. در صورتی که بخواهیم
 چند شرط را مورد بررسی قرار دهیم به صورت زیر عمل نماییم.

شرط اول `if`
 عملیات مورد نظر
 در صورتی که
 شرط دوم `elseif`
 عملیات مورد نظر
`end`

در این حالت اگر شرط اول برقرار نباشد، شرط دوم را بررسی می کنند
 ★ در صورتی که بخواهیم شرط اول صحیح بود یک عملیات و در صورتی که شرط
 صحیح نبود یک عملیات دیگر را اجرا کنیم به صورت زیر عمل نماییم.

شرط اول `if`
 عملیات مورد نظر
 در غیر این صورت `else`
 عملیات مورد نظر
`end`

★ مثال: برنامه‌ای بنویسید که عددی را از ورودی دریافت کند اگر آن عدد بین ۱۰ تا ۲۰ بود کلمه معتبر (Valid) و در غیر اینصورت

کلمه نامعتبر (invalid) را در خروجی چاپ کند
 clear all → تمامی متغیرهای موجود را پاک می‌کند

close all

→ تمامی پنجره‌های اضافی بسته را می‌بندد

clc →

command window را پاک می‌کند

A = input('please enter your number: ');

★ از دستور input برای ورود مقادیر استفاده می‌کنند

if A < 10;

disp('invalid');

elseif A > 20;

disp('invalid');

else

disp('valid');

end

با اجرای برنامه پیام زیر ظاهر می‌شود:

ص ۴

★ تبدیل اعداد باینری به دسیمال و برعکس

bin2dec: تبدیل باینری به دسیمال (اعداد باینری باید حتماً به صورت string باشند)
(نوشته شوند)

dec2bin: تبدیل اعداد دسیمال به اعداد باینری (تبدیل عدد به صفر و یک)

مثال: $\gg A = '1000010'$ ^{string}

$\gg B = \text{bin2dec}(A)$

$B = 44$

$\gg B = 44$

$C = \text{dec2bin}(B)$

$C = 1000010$

★ پردازشی تصویر در متلب:

~~Image~~ Image processing in MATLAB

برای آنالیز و پردازش یک تصویر و شناسایی آن برای نرم افزار

Matlab باید تصویر را قابل فهم برای سیستم تبدیل نمود. تفسیر با/

قابل فهم یک سیستم کامپیوتری اعداد ۰ و ۱ باشد. به همین دلیل

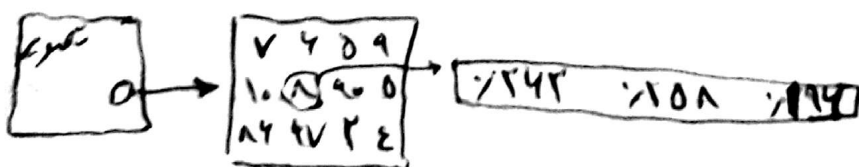
تصاویر به صورت ماتریسی از اعداد تبدیل می شود که نرم افزار تنها
با این اعداد در ارتباط است. برای تصاویر رنگی نیاز به یک ماتریس
برای تصاویر رنگی نیاز به ۳ ماتریس است. و نرم افزار باید هر ماتریس را
جداگانه مورد بررسی قرار دهد.

نرم افزار MATLAB در حالت کلی چهار نمونه تصویر را تولید می کند.

① تصاویر باینری (Binary): تصاویر سیاه و سفید

در این تصاویر آلفا قسمتی از تصویر انتخاب شود
ماتریس آن به صورت مقابل است یعنی فقط دارای صفر و یک است
(اعداد ۰ به معنی رنگ سیاه و اعداد ۱ به معنی رنگ سفید است)

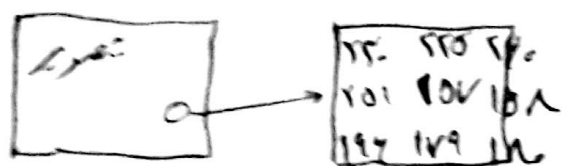
② تصاویر رنگی با شاخص (Indexed):



در این تصاویر آلفا قسمتی از تصویر انتخاب شود و آن ناحیه برای
هر پیکسل یک عدد نسبت داده می شود که این عدد (نمایندگی) ترتیب
ARPACK

سبز، آبی و قرمز است (RGB). پس اگر می‌خواهیم از اعداد انتخاب شود
جمع رنگ ترکیب آن مشخص می‌شود.

اما تصاویر غیر رنگی با ~~رنگ~~ شدت نورانی (Intensity یا Gray) خالص

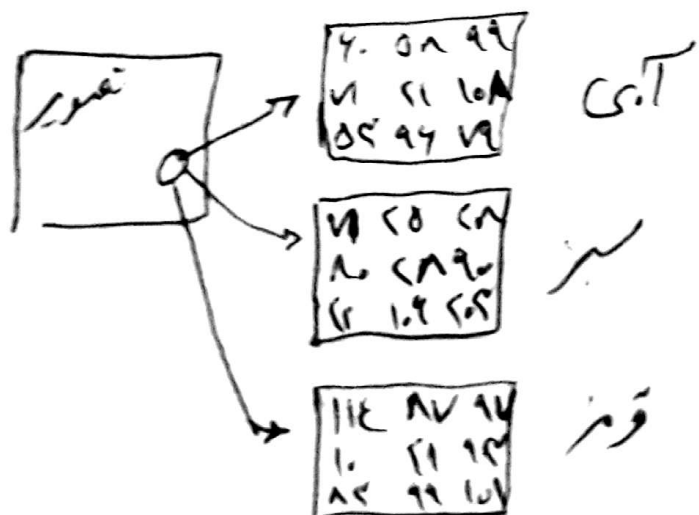


این تصاویر به صورت تک ماتریس
می‌باشد و اعداد نشان‌دهنده شدت

پیکسل نشان‌دهنده روشنایی یا تیرگی آن می‌باشد. در قدیم آرایه‌های این

نوع تصویر عددی بین ۰ تا ۱ بود.

اما تصاویر RGB:



در این تصاویر هر پیکسل از ترکیب ۳ ماتریس آبی، سبز و قرمز ساخته

آمده است.

★ دستورات پردازشی تصویر

★ دستور `imread` : از این دستور برای خواندن این تصویر
(درود تصویر به محیط برنامه افزایر) استفاده می شود.

مثال: `A = imread('نام فایل مسیر جایی تصویر')`

نکته: بهتر است در فایل دستور از (و) استفاده شود (با این کار دستور
انجام گنده ولی عملیات (نوشتن ماتریس تصویر) نشاء داده نمی شود
و در زمان اجرا سریعتر می شود.

و: `p = imread('c:\a.jpg')` مثال

این دستور تصویر با نام `a` که در درایو `c` ذخیره شده را خوانده و در داخل
متغیر `p` قرار می دهد.

★ دستور `iminfo` : اطلاعات کامل را از تصویر می که در مسیر
جاری موجود می باشد را در اختیار کاربر قرار می دهد.

مثال: `iminfo('c:\matlab\alpha.jpg')`

★ این دستور در مورد عکس `alpha` که در پرت `matlab` درایو `c` قرار دارد
یک سری اطلاعات می دهد (فرمت عکس `size` - طول - عرض ...)

* دستور imshow : نمایش تصویر فراخوان شده با ابزار

ویرایشی. با اجرای این دستور عکس در پنجره‌ای باز شده که امکان Zoom

کردن و یا چرخاندن همان تصویر را دارد.

» `A = imread('C:\matlab\alpha.jpg');`

» `imshow(A)`

* دستور `imtool` : نمایش تصویر فراخوان شده با ابزار رنگ پیکسل.

با اجرای این دستور ~~عکس~~ عکس در پنجره‌ای نشان داده شده که علاوه

بر قابلیت Zoom داشتن قابلیت‌های مهم زیر را دارد.

Crop Image : ابزار برای برش قسمت از تصویر. 

Measure distance : ابزار برای اندازه‌گیری فاصله بین دو 

نقطه بر اساس pixel (مثلاً اندازه‌گیری قطر داخل و خارجی اوریجین)

Inspect pixel values : ابزار برای نمایش RGB هر پیکسل

» `A = imread('C:\matlab\alpha.jpg');`

» `imtool(A)`

* دستور `title`: اگر نیاز باشد که در هر پنجره نمایش دهنده تصویر و یا نمایش بردار متنی یا عددی نوشته شود از این تابع استفاده می شود. برای نمایش متن کافی است متن مورد نظر خود را در علامت ' قرار دهید. برای نمایش اعداد باید از دستور `num2str` استفاده شود تا مقدار به رشته تبدیل شود.

نکته: اگر از چند مقدار و متن برای نمایش می خواهیم استفاده نماییم هر کدام را با علامت کاما از هم جدا کرده و همگی را در علامت درآلت [] قرار دهید.

» `A = imread('C:\matlab\1a.jpg');`

» `X = ۲۲۲`

» `imshow(A); title(['image proceeing', num2str(X)])`

نمایش تصویر به این صورت.



* دستور `im2bw`: تولید این دکتور هر نوع تصویر (RGB, Index)

Intensity) را بر تراز به تصویر با این (مردیک) تبدیل کردن

فست این دکتور به صورت `im2bw(pic, n)`

n عددی بین صفر تا ۱ است که تعیین کننده میزان مرز بین کلاه

و سفید می باشد. اگر n نوشته نشود مقدار آن ۰.۵ فرض می شود.

1- clear all

2- close all

3- clc

4- `A = imread('c:\matlab\1a.jpg');`

5- `imshow(A)`

6- `im2bw(A, 0.6);`

7- `imshow(B)`

نکته: با اجرای این برنامه در نهایت تصویر B که یک تصویر با این است نمایش

داده می شود و در خط ۵ نمایش داده نمی شود.

* اجرای مرحله ان برنامه: برای اینکه اجرای برنامه را خطبه خطبه کنترل کنیم.

بر روی خط تیره بعد از عدد سطر که کلیک می کنیم در این حالت یک دایره قرمز
ظاهر می شود. یعنی برنامه تا آنجا اجرا شده و متوقف می شود. برای

ادام اولیه  دکمه Run زده می شود در ادامه می توان

با انتخاب دکمه step  برنامه را خط به خط اجرا نمود.

دکتر graythresh: همیشه می توان یک تصویر را به خوبی به یک

تصویر باینری تبدیل نمود. برای اینکه با تغییر مقدار n در تابع `im2bw`

کیفیت تصویر خروجی تغییر اساسی می کند. برای اینکه بهترین تصویر

را از این تابع بدست آوریم از `graythresh` استفاده می کنیم. این دکتر مقدار

n را نسبت به تصویر اصلی به صورت اترماتیک اعمال می کند و نیاز به n

`clear all`

n نیست.

`close all`

`clc`

`A = imread('c:\matlab\ a.jpg');`

`imshow(A)`

`B = im2bw(A, graythresh(A));`

`imshow(B)`

* دستور `rgb2gray`؛ توسط این دستور تصویر RGB را می توان به تصویر از نوع `Intensity` یا هم / تصویر `Gray` تبدیل کرد.

```
>> A = imread('c:\matlab\ a.jpg');
```

```
>> imshow(A)
```

خاکستری کردن تصویر

```
>> B = rgb2gray(A);
```

```
>> imshow(B)
```

* دستور `dither`؛ توسط این دستور یک تصویر `Intensity` یا هم / تصویر خاکستری را می توان به یک تصویر باینری تبدیل کرد.

```
A = imread('c:\matlab\ a.jpg');
```

```
imshow(A)
```

```
B = rgb2gray(A);
```

```
imshow(B)
```

```
C = dither dither(B);
```

```
imshow(C)
```

ادامہ مطالب کار با دکتورات پردازش تصویر در مجیدہ
نرم افزار متلب بہ کمک کتاب پردازش تصویر در متلب
آقای عبد الرحمن حیدری؛ انتشارات بہ آوران

پدیدآورندگان : نویسنده: عبدالرحمن حیدری
 موضوع : عکس پردازی - روشهای رقمی , مطلب
 ناشر : کلک زرین , به آوران
 محل نشر : تهران
 قطع : وزیری
 نوع کتاب : تالیف
 زبان اصلی : انگلیسی
 قیمت : 60000
 نوع جلد : شومیز
 قطع : وزیری
 تیراژ : 1100
 تعداد صفحات : 288
 تاریخ نشر : 1388/9/29



رده دیویی : 06201367

شابک : 978-600-5516-26-5



Mankan-E11-480P.mp4

Censored

Show all