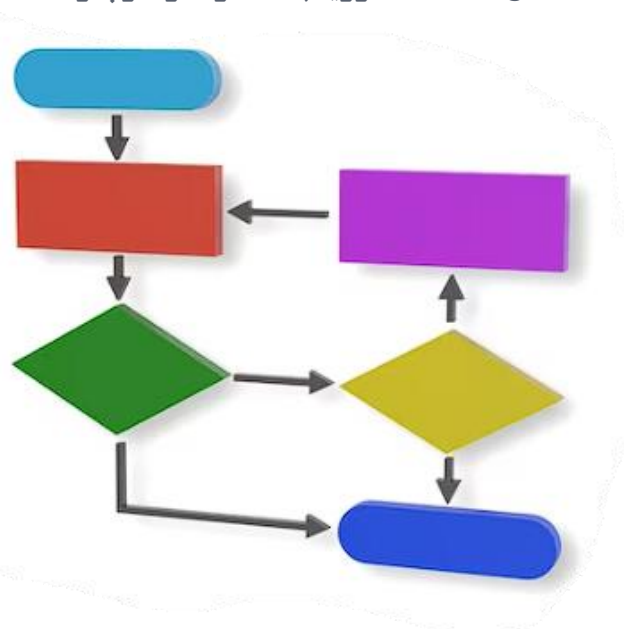


مبانی کامپیوتر و برنامه سازی

اسلاید پنجم

«حل مساله : الگوریتم، عملگرها و فلوچارت»



محمد سعید صفایی صادق

(استفاده از اسلایدها صرفاً برای دانشجویان مجاز می باشد!)

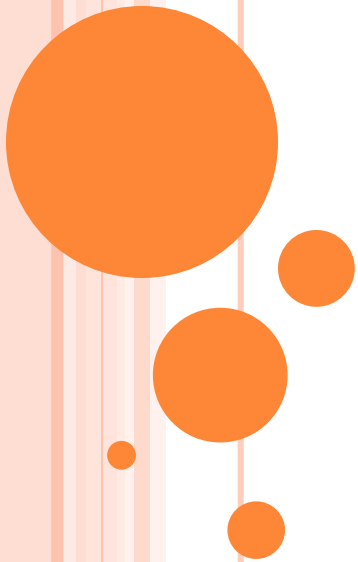
۱۴۰۲

درس

مبانی کامپیوتر

۵

الگوریتم (Algorithm)



الگوریتم

واژه الگوریتم (الخوارزمی)

واژه الگوریتم اقتباسی از نام همه‌چیزدان خوارزمی (الخوارزمی) **Al-Ghoritmi** است.



ابوجعفر محمد بن موسی خوارزمی (۸۵۰-۷۸۰ م.) (۱۸۵-۲۳۳ ه.ق) ریاضیدان، ستاره‌شناس،
جغرافی‌دان، فیلسوف و مورخ و همه‌چیزدان ایرانی.

وی بیشتر عمر خود را در بغداد زیست و از اعضای دارالحکمه بود.

شهرت وی در ریاضیات، به‌ویژه جبر است.

الگوریتم

واژه الگوریتم (الخوارزمی)

رساله‌ای که خوارزمی در قرن ۹ میلادی به عربی نگاشته بود، در قرن ۱۲ به لاتین با نام *Algorithmi de numero Indorum* ترجمه شد؛

یعنی "[کتابی بدست] «الگوریتمی» در مورد اعداد هندی"، که «الگوریتمی» نام الخوارزمی بود که مترجم در تبدیل به لاتین نام وی را جلوی نام اصلی کتاب آورده بود.

در قرن ۱۳ میلادی واژه الگوریسموس *algorismus* به معنای «سیستم شمارش عربی (دهدهی)» (یعنی اعداد ۱ تا ۹ به علاوه صفر، و نیز مفهوم اعشار) بود؛

معنای دیگر الگوریسم «حساب کردن با کمک اعداد عربی» است؛

در قرن ۱۹ این کلمه در فرانسوی به *algorithme* تغییر شکل پیدا کرد، البته معنایش ثابت ماند.

طولی نکشید که این کلمه به شکل *algorithm* وارد زبان انگلیسی شد؛ و در اواخر قرن ۱۹ میلادی بود که معنای عام‌تر امروزی‌اش را یافت.

الگوریتم چیست ؟

بشر در طول زندگی خود با مشکلات بسیاری مواجه می‌شود که ممکن است بسیاری از آنها بارها تکرار شوند.

مسائلی از قبیل راه رفتن غذا، خوردن خوابیدن و غیره که بشر تقریباً هر روز آنها را پیش روی خود دارد.

همه این مسائل نیاز به روشی برای حل کردن دارند مثلاً راه رفتن باید با ترتیب خاصی و مراحل معینی انجام شود تا مسئله راه رفتن برای بشر حل شود.

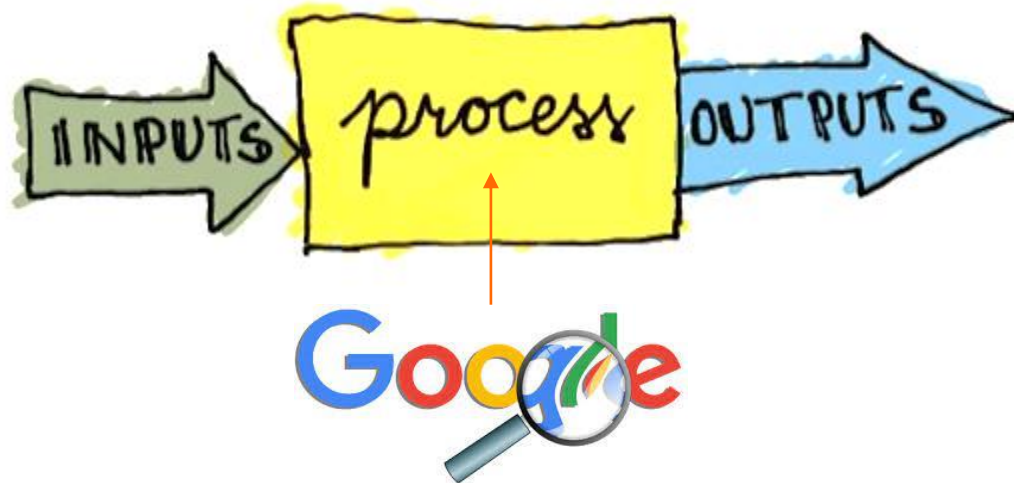
اصطلاحاً روش انجام کار یا حل مسئله را الگوریتم آن مسئله می‌نامند.

ما اغلب برای حل مشکلات به دنبال ساده‌ترین و سریع‌ترین راه‌حل‌ها هستیم.

ثبت و دسته‌بندی راه‌حل‌های مشخص کمک می‌کند که بتوانیم سریع‌تر و مطمئن‌تر با مسائل تکراری برخورد کنیم.

ما به این راه‌حل‌های تست شده و مطمئن، الگوریتم می‌گوییم.

الگوریتم چیست ؟



برای مثال :

الگوریتم موتور جستجو گوگل به طور ساده این گونه است که عبارت تایپ شده شما را به عنوان ورودی دریافت کرده،

آن را در پایگاه داده‌های خود جستجو می کند

و سپس صفحات وب مربوطه را پیدا کرده و در خروجی به شما نشان می دهد.

این روند کلی از ایجاد سؤال تا رسیدن به پاسخ یک الگوریتم محسوب می شود.

الگوریتم

مثال الگوریتم در دنیای واقعی :

برای آماده کردن یک نوشیدنی مانند چای یا قهوه می توان به صورت زیر یک الگوریتم طراحی کرد:

مرحله اول: شروع

مرحله دوم: آب برای درست کردن قهوه در ظرف جوشانده می شود.

مرحله سوم: پودر قهوه به آب جوش اضافه می شود.

مرحله چهارم: شکر به قهوه اضافه می شود.

مرحله پنجم: شیر همراه با شکر به قهوه ساخته شده در مرحله قبل اضافه می شود.

مرحله ششم: همه مواد این قهوه به مدت ۲ دقیقه جوشانده می شوند.

مرحله هفتم: قهوه در فنجان ریخته می شود.

مرحله هشتم: پایان



مثال الگوریتم در دنیای واقعی :

در بخش پایین الگوریتم استفاده از تلفن سکه ای را شرح میدهیم:



- ۱- شروع کن
- ۲- گوشی تلفن را بردار
- ۳- یک سکه در تلفن بنداز
- ۴- صبر کن تا صدای بوق آزاد را بشنوی
- ۵- شماره مورد نظر را بگیر
- ۶- صحبت کن
- ۷- گوشی را در محل قرار بده
- ۸- پایان عملیات

الگوریتم

ویژگی های الگوریتم :

الگوریتم مجموعه ای از دستورالعملها برای حل مسئله میباشد که مراحل زیر را باید دارا باشد:

۱- تعریف دقیقی از مساله

۲- تعیین کلیه پارامترها یا عوامل موجود در مساله

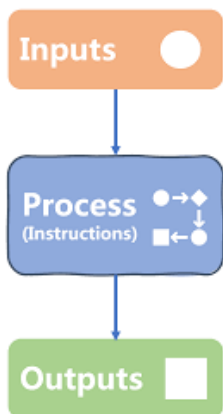
۳- ورودی و خروجی های مساله

۴- انتخاب یک راه حل مناسب برای حل مساله

مراحل الگوریتم برای حل یک مسئله:

اصطلاحاً پیدا کردن راه حل برای مساله، طراحی الگوریتم برای آن مسئله

نامیده می شود. در طراحی الگوریتم معمولاً سه مرحله زیر را از هم جدا کنند



۱- خواندن داده ها (ورودی)

۲- انجام محاسبات (پردازش)

۳- نمایش اطلاعات خروجی

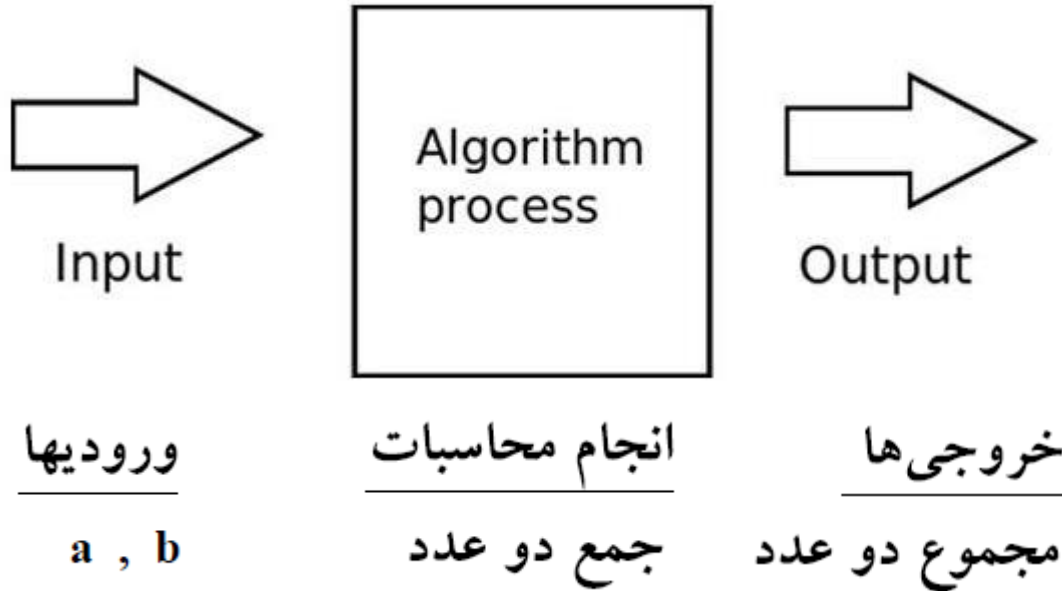
تعریف الگوریتم :

الگوریتم یعنی برنامه و طرحی که صفر تا صد کاری را بتوان مبتنی بر آن انجام داد. الگوریتم دارای خروجی است و وقتی آن را برقرار می‌دانیم که اجرای آن در زمان محدود به اتمام برسد.

در علوم کامپیوتر، الگوریتم **Algorithm** را مجموعه‌ای از گام‌ها و فرآیندهای پی‌درپی برای حل مساله یا انجام محاسبات تعریف کرده‌اند.

الگوریتم

مثال : الگوریتمی بنویسید که دو عدد از ورودی دریافت کرده مجموع دو عدد را محاسبه و چاپ نماید.



۱- شروع

۲- a و b را بخوان

۳- مجموع a و b را محاسبه و در `sum` قرار بده.

۴- `sum` را در خروجی چاپ کن

۵- پایان

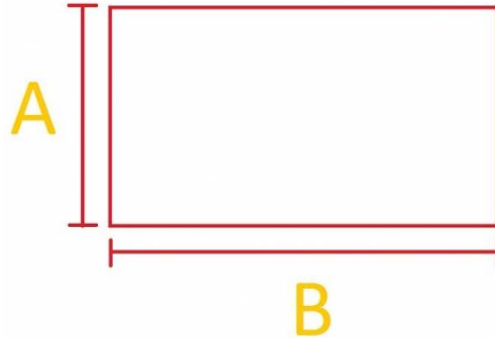


sum

دیکشنری انگلیسی به فارسی

جمع کردن، حاصل جمع، مجموع
مبلغ، حاصل جمع، روی هم، خلاصه، مختصر،
حساب کردن، باهم جمع کردن، مختصر و موجز کردن،
خلاصه نمودن [کامپیوتر]: حاصل، جمع .

مثال : الگوریتمی بنویسید که مساحت مستطیل را محاسبه کند



Input	Process	Output
<u>a,b</u> (طول و عرض)	$c=a*b$ (فرمول مساحت)	c

۱- شروع

۲- a و b را بخوان

۳- حاصلضرب a و b را محاسبه و در c قرار بده.

۴- c را در خروجی چاپ کن

۵- پایان

الگوریتم

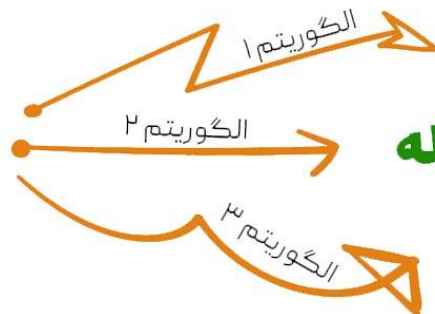
نکته:

برای حل یک مساله گاهی چندین راه حل پیشنهاد میگردد. که جواب همگی صحیح می باشد.

مثال : الگوریتمی بنویسید که مساحت مستطیل را محاسبه کند

۳	۲	۱
۱- شروع ۲- عدد ۱ را از کاربر بگیر ۳- عدد دیگر را از کاربر بگیر ۳- عدد اول را در دومی ضرب کن ۴- حاصل مرحله ۴ را چاپ کن ۵- پایان	۱- شروع ۲- عدد ۲ را از کاربر بگیر ۳- اولی را در دومی ضرب کن ۴- حاصل مرحله ۳ را چاپ کن ۵- پایان	۱- شروع ۲- a و b را بخوان ۳- حاصل ضرب a و b را محاسبه و در c قرار بده. ۴- c را در خروجی چاپ کن ۵- پایان

مسئله



پاسخ مسئله

الگوریتم

چگونه الگوریتم بنویسیم:

برای حل کردن یک مسئله احتیاج به پیدا کردن الگوریتم ها برای رسیدن به پاسخ داریم. از بین الگوریتم ها، می بایست بهترین الگوریتم را از نظر بهینه بودن انتخاب کنیم. در برنامه نویسی نیز شما به دفعات زیاد با این رو مسائل و الگوریتم ها روبرو می شوید و با تبدیل الگوریتم انتخاب شده به کد، کار خود را به اتمام می رسانید. وقتی شما تصمیم می گیرید که یک برنامه نویس شوید باید بتوانید در مواجهه با مسائل و چالش های پیش رویتان، الگوریتم های متعددی را پیدا کنید.

برای الگوریتم نویسی نیاز به **شناخت مساله و پیدا کردن راه حل** داریم

شناخت مساله:

خوب می‌دونیم که اولین قدم از حل یک مسئله ، شناختن اون مسئله هستش و برای شناخت یک مسئله باید به داده های مسئله دقت کنیم !

مثال : اگر سارا ۲ سال سن داشته باشد ، او چند ماه عمر کرده است ؟!

در این مثال به وضوح می‌تونیم ببینیم که سن سارا ، یک داده مسئله هستش و تویه شناخت مسئله خیلی میتونه کمکون کنه !

بعد از اینکه داده های یک مسئله رو در نظر گرفتیم ، باید صورت مسئله هم بتونیم درک کنیم و اصلا بفهمیم این مسئله از ما چه چیزی رو می‌خواود و دنبال چیه ؟!

به مثال دقت کنید ، متوجه میشید که مسئله بازم سن سارا رو می‌خواود اما به ماه می‌خاد و یه جورایی می‌فهمیم که صورت مسئله ، تبدیل سن سارا از سال به ماه هستش !

پیدا کردن راه حل :

وقتی مسئله رو خوب شناختیم نوبتش رسیده با استفاده از شناختی که ازش داریم ، هر تعداد راه حل که به ذهنمون میاد رو تویه ذهنمون داشته باشیم یا جایی یادداشت کنیم .

از بین راه حل های ارائه شده ، قطعا بخشی از اونا منطقی هستند و بخشی هم غیر منطقی و وظیفه ما اینه که منطقی ترین و کوتاه ترین راه حل رو انتخاب کنیم.

مثلا در مثال قبل:

کلید حل مساله تبدیل سال به ماه است.

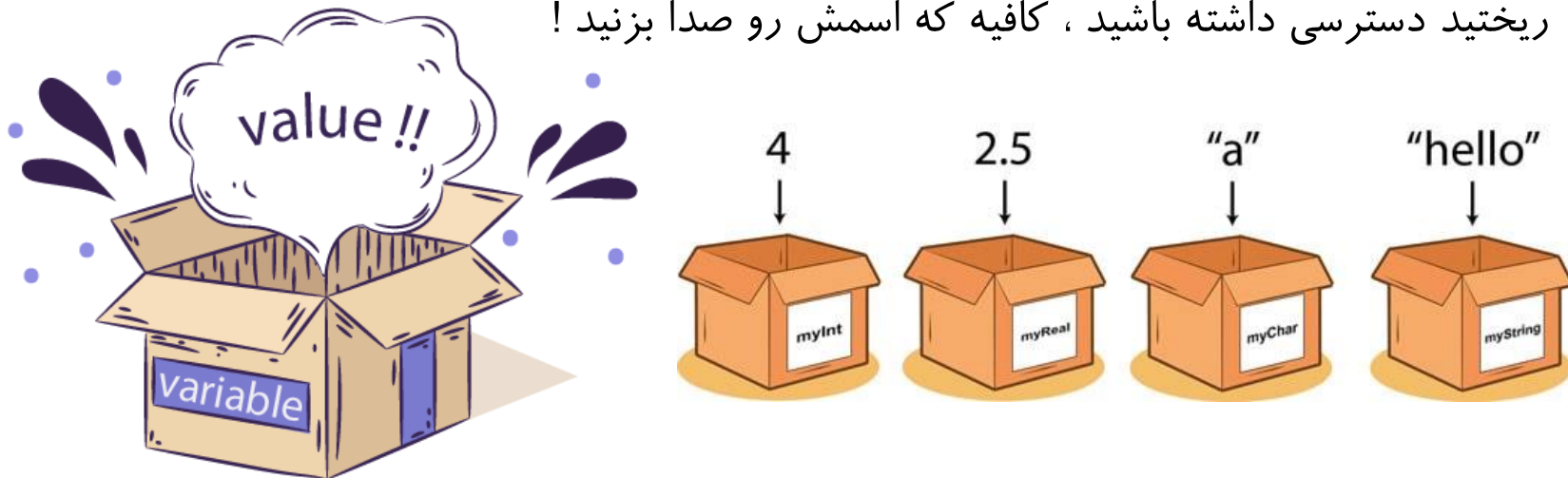
هر سال ۱۲ ماه است.

پس فقط کافیست سن سارا را به سال ضرب در ۱۲ کنیم تا تعداد ماه های عمر او نمایش داده شود.

مفهوم متغیر یا Variable :

یک خونه یا جعبه رو تصور کنید ، میدونیم تو هر دو اونها یه فضایی هستش که میتونیم توش یه چیزی قرار بدیم و برامون نگه داره !

حالا اون جعبه رو داخل کامپیوتر تصور کنید ، میتونه براتون یه حافظه ای اختصاص بده که تو اون حافظه ، چیزی رو که شما بخواید رو قرار میده و شما هر وقت بخواید به چیزی که توش ریختید دسترسی داشته باشید ، کافیه که اسمش رو صدا بزنیند !



مثال : عدد ۵ رو داخل جعبه ای به نام X میریزیم ، حالا هر وقت بخوایم به عدد ۵ دسترسی داشته باشیم کافیه X رو صدا بزنینم چرا که X همون عدد ۵ هست و با هم برابر هستند ، در حقیقت X یک متغیری هست که درون حافظه خودش عدد ۵ رو قرار داده !

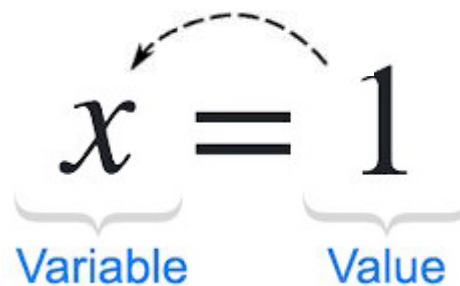
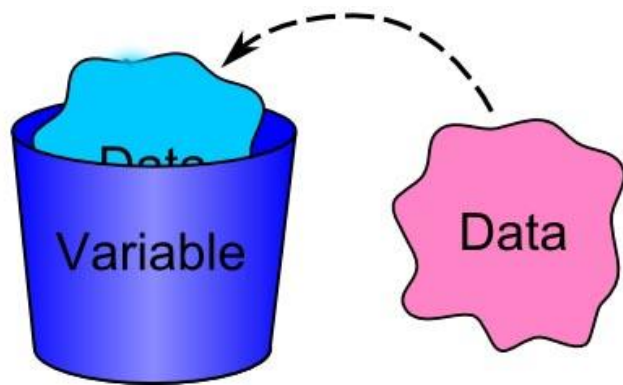
عملگر مساوی (=) :

این عملگر هم مثل عملگر مساوی ریاضیاته اما با این تفاوت که از این عملگر برای نشون دادن برابری دو عبارت استفاده نمیشه.

بلکه از این عملگر در برنامه نویسی برای قرار دادن مقدار حاصل از یک عبارت در یک متغیر استفاده میشه. (دستورات انتسابی)

برای مثال اگر بخواهیم دو عدد رو با هم جمع کنیم و در متغیر X بریزیم باید بنویسیم $X = 1 + 3$.

نکته ای که درباره ی این عملگر باید بهش دقت بشه اینه که همیشه سمت چپ این عملگر باید یک متغیر (یا هر مکانی از حافظه) قرار بگیره تا مقدار حاصل از عبارت در سمت راست داخل اون متغیر در سمت چپ ریخته بشه.



الگوریتم

مثال : الگوریتمی بنویسید که میانگین ۳ عدد را محاسبه و چاپ کند.

۱- شروع

۲- a و b و c را بخوان

۳- حاصل جمع a و b و c را محاسبه و در متغیر sum قرار بده.

۴- sum را تقسیم بر ۳ کن و در متغیر avr قرار بده

۵- avr را چاپ کن

۶- پایان



average

دیکشنری انگلیسی به فارسی

متوسط معدل، حد وسط، میانه، متوسط، درجه عادی، میانگین، حد وسط (چیزی را) پیدا کردن، میانه قرار دادن، میانگین گرفتن، رویهم رفته، بالغ شدن [الکترونیک]: متوسط [عمران و معماری]: متوسط [عمران و معماری]: میانگین

الگوریتم

مثال : الگوریتمی بنویسید که ۲ عدد را دریافت و محاسبه شود کدام عدد بزرگتر است.

۱- شروع

۲- a و b را بخوان

۳- اگر a از b بزرگتر بود آنگاه مقدار a را در متغیر \max قرار بده

۴- در غیر اینصورت b را در متغیر \max قرار بده

۵- \max را چاپ کن

۶- پایان

مثال : الگوریتمی بنویسید که ۲ زاویه را دریافت و محاسبه شود آیا متمم هستند یا خیر؟

۱- شروع

۲- a و b را بخوان

۳- اگر $a+b=90$ بود مقدار "yes" را چاپ کن

۴- در غیر اینصورت "no" را چاپ کن

۵- پایان

الگوریتم

مثال : الگوریتمی بنویسید که ۱ عدد را از ورودی دریافت کند نشان دهد زوج است یا فرد؟

۱- شروع

۲- a را بخوان

۳- عدد a را بر ۲ تقسیم کن

۴- اگر باقیمانده ۰ بود چاپ کن "عدد زوج است".

۵- در غیر اینصورت چاپ کن "عدد فرد است".

۶- پایان



even

دیکشنری انگلیسی به فارسی

عدد زوج هموار، صاف، مسطح، تراز، مساوی، هموارکردن، صاف کردن، وارپز کردن، حتی، هم، درست، اعداد جفت [عمران و معماری]: یکدست کردن [عمران و معماری]: یکنواخت کردن [عمران و معماری]: همواری [عمران و معماری]: مسطح [عمران و معماری]: هموار

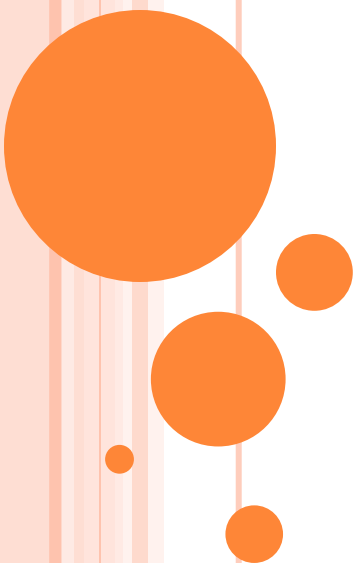


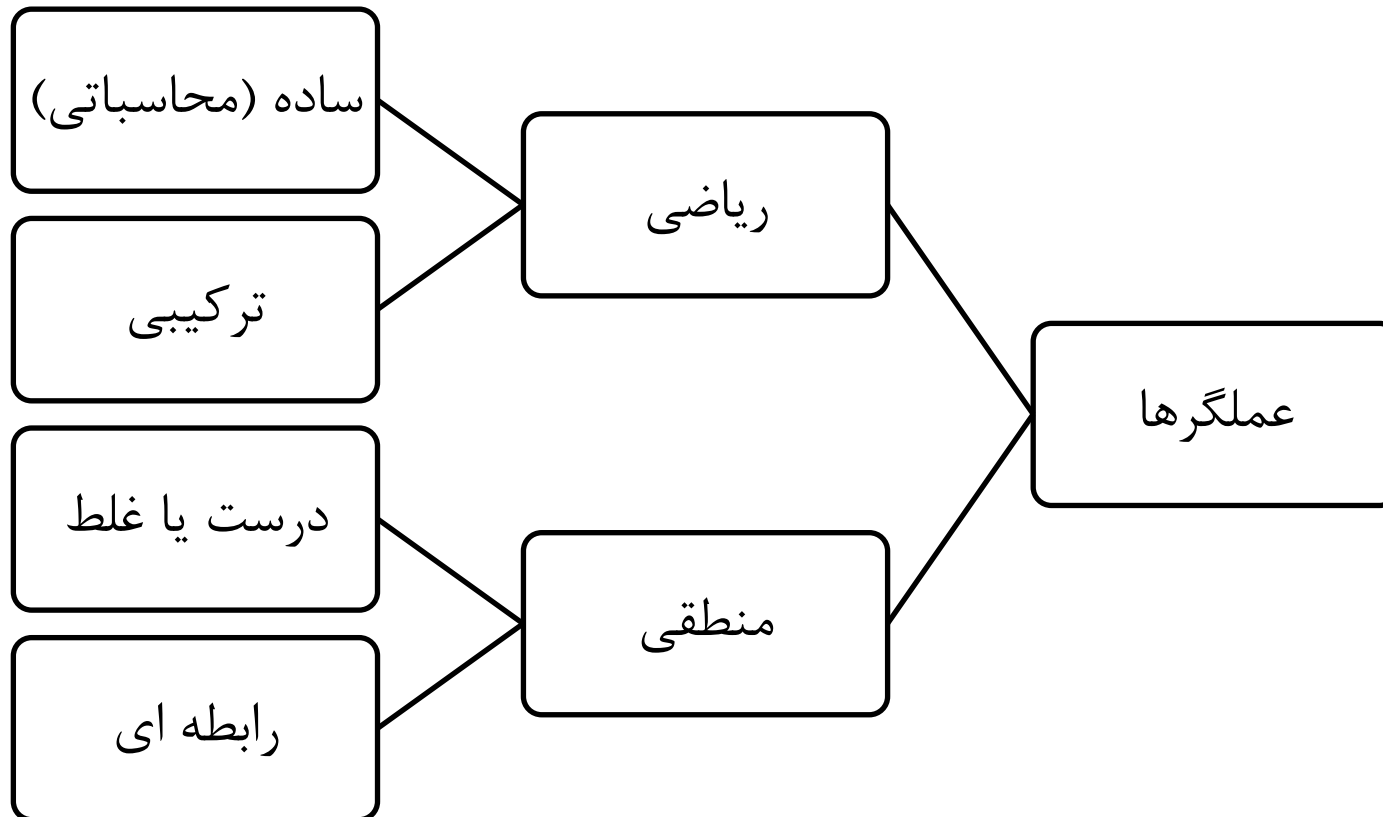
odd

دیکشنری انگلیسی به فارسی

عدد فرد سوگند ملایم، بخدا، طاق، تک، عجیب و غریب، آدم عجیب، نخاله

انواع عملگرها





عملگرها

۱- عملگرهای محاسباتی:

در الگوریتم نویسی، یکی از پرکاربردترین نوع داده ها انواع عددی هستند و قاعدتاً انجام عملیات مقدماتی ریاضیاتی مانند جمع، تفریق، ضرب، تقسیم و ... نیز روی این نوع داده ها به دفعات صورت می گیرد.

عملگرهای محاسباتی که تقریباً شبیه به همان نمادهایی هستند که ما در ریاضیات برای انجام این عملیات از آن ها استفاده می کنیم.

عملگر	نام	مثال
-	تفریق و منهای یکانی	X-Y یا -Z
+	جمع	X+Y
*	ضرب	X*Y
/	تقسیم	X/Y
%	باقی مانده تقسیم	X%Y
--	کاهش DECREMENT	X- یا -X
++	افزایش INCREMENT	++X

عملگرها

۲- عملگرهای رابطه ای:

عملگرهای رابطه ای (شرطی) دو مقدار رو باهم مقایسه می کند و رابطه ی دو مقدار رو تعیین، و نتیجه ای که از اجرای این عملگرها به دست می آید همیشه درست یا نادرست می باشد.

عملگر	نام	مثال
>	بزرگتر	$x > y$
>=	بزرگتر یا مساوی	$x >= y$
<	کوچکتر	$x < y$
<=	کوچکتر یا مساوی	$x <= y$
==	متساوی	$x == y$
!=	نامساوی	$x != y$

عملگرها

۳- عملگرهای منطقی (درست و غلط): Logical Operators
عبارات منطقی دارای دو ارزش درستی و نادرستی هستند.

عملگر	نام	مثال
!	نقض (not)	$!y$
&&	و (and)	$x > y \ \&\& \ m < p$
	یا (or)	$x > y \ \ m < p$



T
true

دیکشنری انگلیسی به فارسی

راست، پارچا، ثابت، واقعی، حقیقی، راستگو، خالصانه، صحیح، ثابت یا حقیقی کردن، درست، راستین، فربور درست، صحیح، واقعی [الکترونیک]: درست، صحیح



F
false

دیکشنری انگلیسی به فارسی

دروغ، کذب، کاذبانه، مصنوعی، دروغگو، ساختگی، نادرست، غلط، قلبی، بدل [الکترونیک]: نادرست

عملگرها

جدول درستی عملگرهای منطقی Logical Operators:

p	q	p && q	p q	! p	! q
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	F
F	F	F	F	T	T

- نتیجه عملگر ! وقتی درست است که عملوندهای آن دارای ارزش نادرستی باشد.
- نتیجه عملگر && وقتی درست است که هر دو عملوند ارزش درستی داشته باشند .
- نتیجه عملگر || تا وقتی نادرست است که هر دو عملوند ارزش نادرستی داشته باشند.

عملگرها

&& رو با این مثال دیگه فراموش نمیکنی!

AND: فرض کن p نصف مسیر رسیدن به خانه q نیمه دیگر مسیر است.

$T \& \& T$: نصف اول مسیر سالم است و نصف دیگرش هم سالم

پس به مقصد میرسی

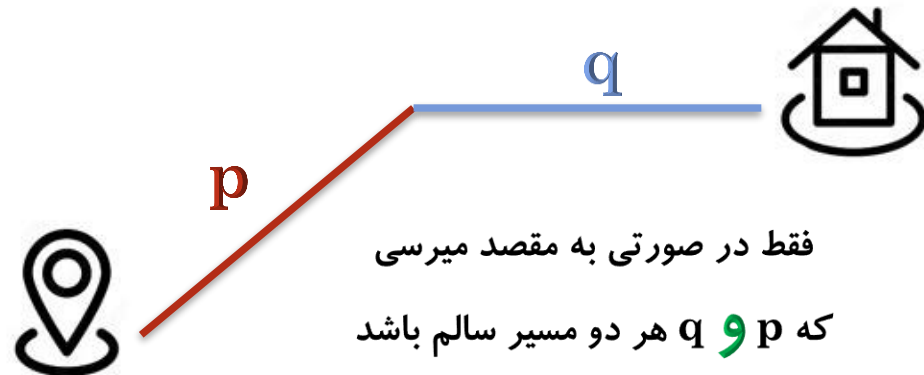
$T \& \& F$: نصف مسیر سالم است و نصف دیگرش خراب

$F \& \& T$: نصف اول مسیر خراب است و نصف دیگرش سالم

$F \& \& F$: نصف اول مسیر خراب است و نصف دیگرش هم خراب

پس به مقصد نمیرسی

p	q	$p \& \& q$
T	T	T
T	F	F
F	T	F
F	F	F



عملگرها

|| رو با این مثال دیگه فراموش نمیکنی!

OR : فرض کن p یک راه رسیدن به خانه، همچنین p مسیری دیگر برای رسیدن به خانه است.

$T \parallel T$: مسیر اول سالم است و مسیر دیگر هم سالم

$F \parallel T$: مسیر اول سالم است و مسیر دیگر خراب

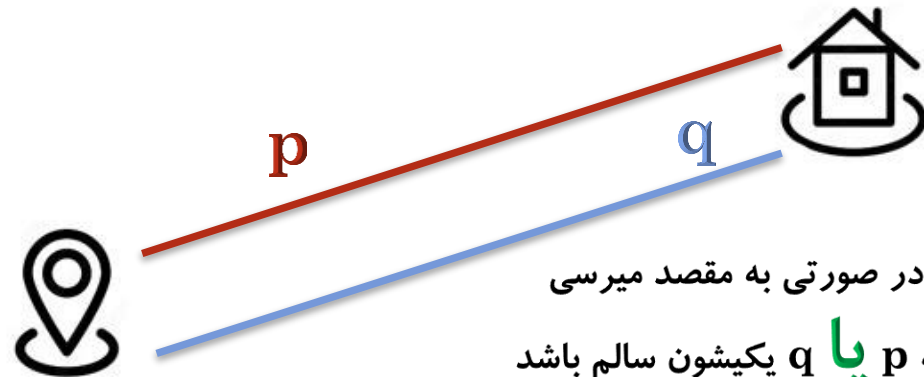
$T \parallel F$: مسیر اول خراب است و مسیر دیگر سالم

پس به مقصد میرسی

$F \parallel F$: مسیر اول خراب است و مسیر دیگرش هم خراب

پس به مقصد نمیرسی

p	q	$p \parallel q$
T	T	T
T	F	T
F	T	T
F	F	F



عملگرها

* ارزش گذاری عبارات :

$$\overset{\mathbf{F}}{(4==5)} \ \&\& \ \overset{\mathbf{T}}{(4!=3)} \longrightarrow \mathbf{F}$$

$$\overset{\mathbf{T}}{(3==3)} \ || \ \overset{\mathbf{F}}{!(\overset{\mathbf{T}}{(3<6)})} \longrightarrow \mathbf{T}$$

$$\overset{\mathbf{T}}{(!0)} \ \&\& \ \overset{\mathbf{T}}{4} \longrightarrow \mathbf{T}$$

T ارزش true برابر با 1 است
F ارزش false برابر با 0 است
T ارزش هر عدد غیر صفر برابر با 1 است

تمرین کلاسی : ارزش عبارات زیر را به دست آورید:

فقط یک **F** داشته باشد کل عبارت **F** میشود
 $a \ \&\& \ b \ \&\& \ c \ \&\& \ d$

فقط یک **T** داشته باشد کل عبارت **T** میشود
 $a \ || \ b \ || \ c \ || \ d$

$$\overset{\mathbf{T}}{(0 \ || \ 1)} \ \&\& \ \overset{\mathbf{T}}{(7<9)} \longrightarrow \mathbf{T}$$

$$\overset{\mathbf{F}}{!9} \ \&\& \ \overset{\mathbf{T}}{(11>=10)} \longrightarrow \mathbf{F}$$

$$\overset{\mathbf{T}}{!(10 \ \&\& \ 0)} \ || \ \overset{\mathbf{F}}{!(0>-3)} \longrightarrow \mathbf{T}$$

عملگرها

۴- عملگرهای ترکیبی:

از ترکیب عملگرهای محاسباتی و علامت = ، مجموعه دیگری از عملگرها ایجاد می شود که عمل محاسباتی و انتساب را انجام می دهند.

معادل	مثال	نام	عملگر
$x = x + y$	$x += y$	انتساب جمع	$+=$
$x = x - y$	$x -= y$	انتساب تفریق	$-=$
$x = x * y$	$x *= y$	انتساب ضرب	$*=$
$x = x / y$	$x /= y$	انتساب تقسیم	$/=$
$x = x \% y$	$x \% = y$	انتساب باقی مانده تقسیم	$\% =$

عملگرها

معنای علامت & (امپرسند Ampersand)

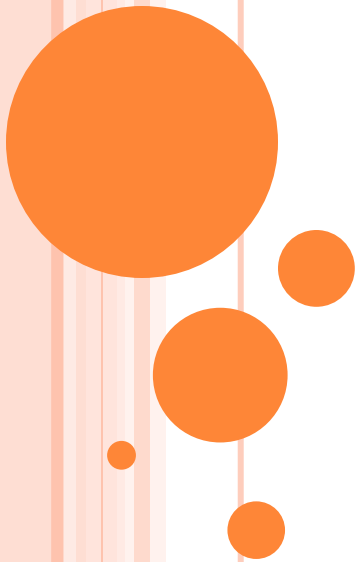
علامت & ریشه در زبان لاتین دارد.

نویسندگان های عصر امپراطوری روم برای اینکه سریع تر بنویسند حروف لاتین را به هم می چسبانند (یا به تعبیری شکسته نویسی می کردند).

در زبان لاتین کلمه ی ربط «و» می شود et .



فلوچارت (FlowChart)



فلوچارت Flow-Chart چیست؟

پس از آنکه مفهوم یک الگوریتم را بدانیم، ساده‌سازی آن اهمیت زیادی دارد. اینکه بتوان مراحل را گام به گام ترسیم کرد و حتی به شکلی ساده‌تر، آن را به دیگران توضیح داد.

برای فهم بهتر یک الگوریتم برنامه‌نویسی و نحوه عملکرد آن می‌توان از فلوچارت FlowChart استفاده کرد.

فلوچارت، نمایشی گرافیکی از فرایند یک برنامه است و از مجموعه شکل‌های قراردادی تشکیل شده که به کمک آن‌ها می‌توان دستورات را به ترتیب ترسیم نموده و روند یک الگوریتم را ساده‌سازی کرد.

کشیدن فلوچارت مربوط به قبل از شروع برنامه‌نویسی است و وابستگی به نوع زبان برنامه‌نویسی ندارد.

فلوچارت‌ها در تعیین روند اجرای دستورات و تعیین ورودی و خروجی برنامه نقش مهمی دارند.



تفاوت الگوریتم و فلوچارت در چیست؟

فلوچارت	الگوریتم
طرحی گرافیکی از فرآیند حل مسئله است.	فرآیندی برای حل مسئله است
فرآیند حل مسئله به صورت نمادهای بصری نمایش داده شده است.	فرآیند حل مسئله به صورت قدم به قدم توضیح داده شده است.
درک فلوچارت ها بسیار ساده است.	الگوریتم ها پیچیده هستند و درک آن ها دشوار است.
عملیات رفع خطا یا debug در فلوچارت ها آسان است.	عملیات رفع خطا یا debug در الگوریتم ها دشوار است.
راه حل به صورت بصری و گرافیکی توضیح داده شده است.	راه حل با زبانی ساده و طبیعی توضیح داده شده است.
حل مشکلات پیچیده با فلوچارت کار سختی است.	برای حل مشکلات پیچیده، انتخاب بهتری است.
ایجاد یک فلوچارت ساده است و زمان زیادی نخواهد برد.	ساخت و تعریف یک الگوریتم جدید زمان زیادی می برد.

فلوچارت

۶ نماد اصلی فلوچارت Flow-Chart

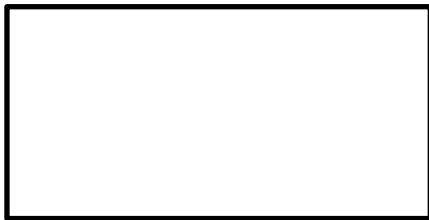
۱- **Terminator** : که شروع و پایان فرایند را مشخص کرده و به شکل بیضی نشان داده می‌شود.

هر فلوچارت یک **ترمیناتور** برای شروع و یکی برای پایان دارد.  

۲- **دریافت اطلاعات ورودی یا چاپ اطلاعات خروجی Data** : ورودی‌ها به شکل متوازی الاضلاع نشان داده می‌شوند.

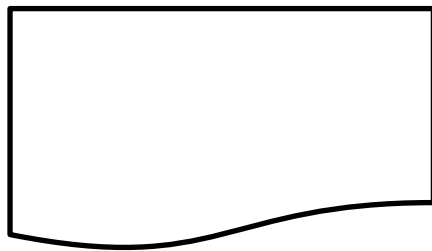
۳- **فرایند Process (محاسبات)** : به شکل مستطیل که برای نمایش دستورات معمولی مانند دستورات محاسباتی و انتسابی (صفحه ۱۷) استفاده می‌شود.



فلوچارت

۶ نماد های اصلی فلوچارت Flow-Chart

۴- سند Document : برای نمایش خروجی و گزارش نهایی از چهارضلعی منحنی استفاده می شود.

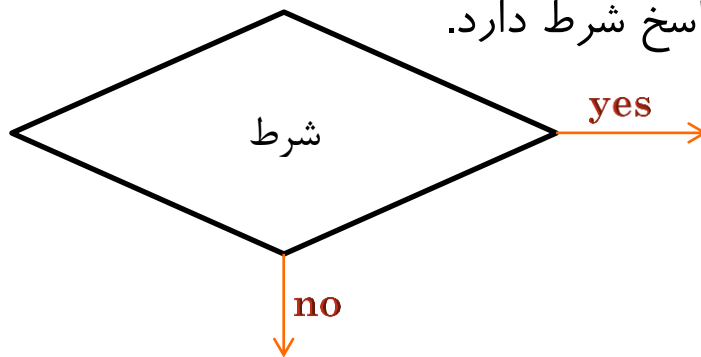


میتوان برای چاپ اطلاعات خروجی از روش ۲ یعنی Data هم استفاده کرد. (با استفاده از متوازی الاضلاع)

۵- ترتیب : برای مشخص کردن ترتیب فرایندها از فلش Arrow استفاده می شود.



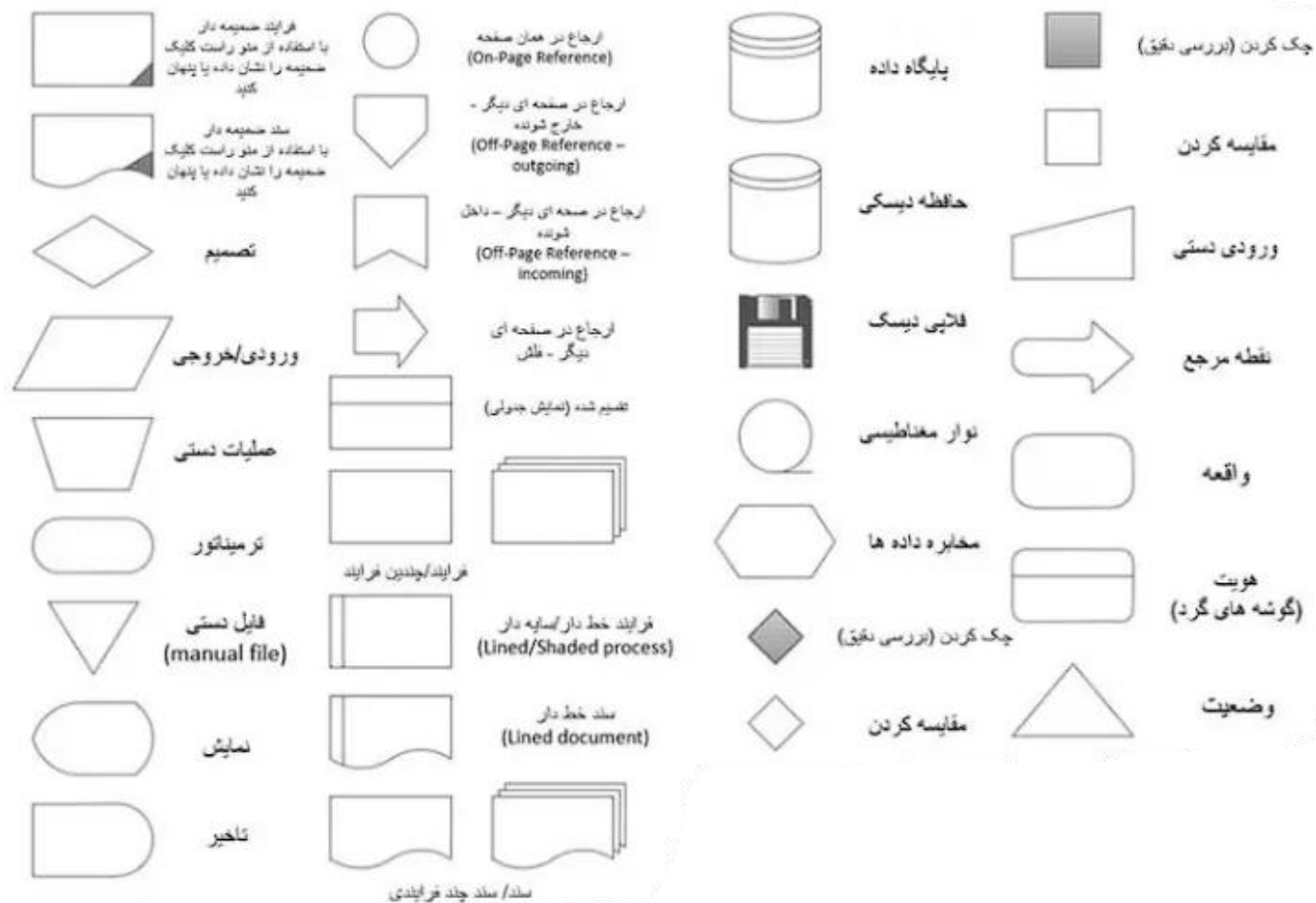
۶- تصمیم Decision : تصمیم گیری برای برقراری شرط است و با لوزی نشان داده می شود. این مرحله دو خروجی متفاوت وابسته به پاسخ شرط دارد.



* علاوه بر این ۶ مورد، نمادهای دیگری در فلوچارت وجود دارد

که در صورت نیاز به توضیح آنها خواهیم پرداخت.

سایر نماد های فلوچارت Flow-Chart



استفاده از Trace : (ردیابی)

بهتر است همزمان با نوشتن الگوریتم آنرا Trace کنیم تا در جریان الگوریتم باشیم و اگر الگوریتم ما مشکلی داشت قبل از اجرا با کامپیوتر آنرا رفع کنیم.

همچنین با Trace می توان مراحل بعدی الگوریتم را حدس زد و در صورت درستی به حل مساله اضافه کرد.

Trace

یعنی آزمودن دستی، همراه با داده های نمونه، جهت ردیابی صحت اطلاعات خروجی



trace

دیکشنری انگلیسی به فارسی

رد، ردیابی کردن، رسم کردن اثر، نشان، رد پا، جای پا، مقدار ناچیز، ز ترسیم، رسم، ترسیم کردن، ضبط کردن، کشیدن، اثر گذاشتن، دنبال کردن، پی کردن، پی بردن به [کامپیوتر]: پیگیری، ردیابی، دنبال کردن ۱. (در يك برنامه ی نقاشی) ایجاد مسیر یا شما در اطراف منحنی های يك شكل طرح بپی ...

متغیر ۱	متغیر ۲	...	N مقدار	خروجی (ها)
مقدار ۱	مقدار ۱	...	مقدار ۱	؟
مقدار ۲	مقدار ۲	...	مقدار ۲	؟
...	؟
N مقدار	N مقدار	...	N مقدار	؟

استفاده از Trace

برای Trace برنامه جدولی میکشیم ، متغیرها و خروجی یا خروجی های برنامه را در سطر اول می چینیم ، و در سطر های بعدی مقادیر آنها را می گذاریم.

خروجی (ها)	N مقدار	...	متغیر ۲	متغیر ۱
؟	مقدار ۱	...	مقدار ۱	مقدار ۱
؟	مقدار ۲	...	مقدار ۲	مقدار ۲
؟
؟	N مقدار	...	N مقدار	N مقدار

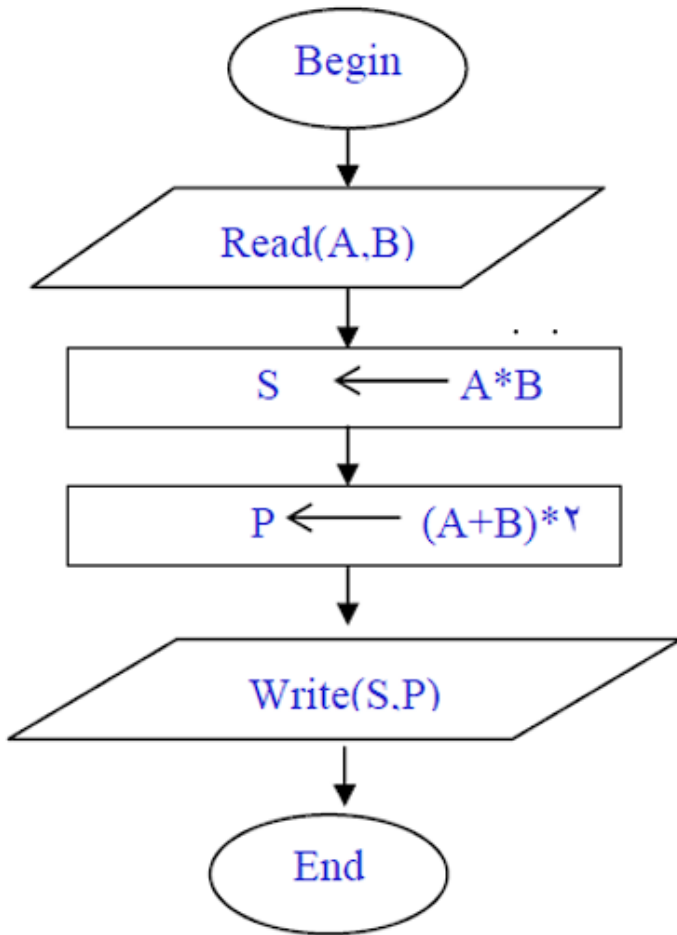
مرحله به مرحله جلو میرویم و همینطور مقادیر آنها را با توجه به روند برنامه میگذاریم... تا اینکه به خروجی برسیم .

خروجی بدست آمده را با خروجی مورد انتظار خود مقایسه می کنیم.

اگر درست بود که یعنی ، الگوریتم درست است و اگر نبود الگوریتم را اصلاح میکنیم تا به جواب برسیم.

فلوچارت

مثال : الگوریتمی بنویسید که مساحت و محیط مستطیل را محاسبه کند، سپس فلوچارت آن را رسم کنید (با استفاده از جدول Trace چک کنید):



• مساحت مستطیل : طول ضرب در عرض

• محیط مستطیل : (طول+عرض) ضرب در ۲

۱- شروع

۲- a و b را بخوان

۳- حاصل ضرب a و b را محاسبه و در S قرار بده.

۴- حاصل جمع a و b را در عدد ۲ ضرب کن و در P قرار بده.

۵- S و P را در خروجی چاپ کن

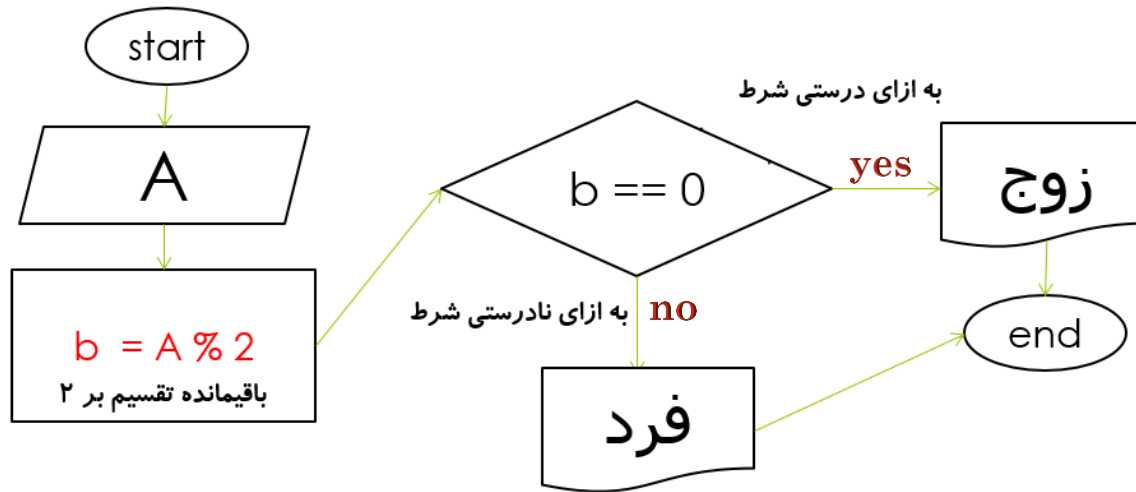
۶- پایان

Trace

A	B	S	P
2	3	6 ✓	10 ✓
1	5	5 ✓	12 ✓
3	4	12 ✓	14 ✓

فلوچارت

مثال : فلوچارتی رسم کنید که عددی را از ورودی دریافت کرده، فرد یا زوج بودن آن را تشخیص دهد.

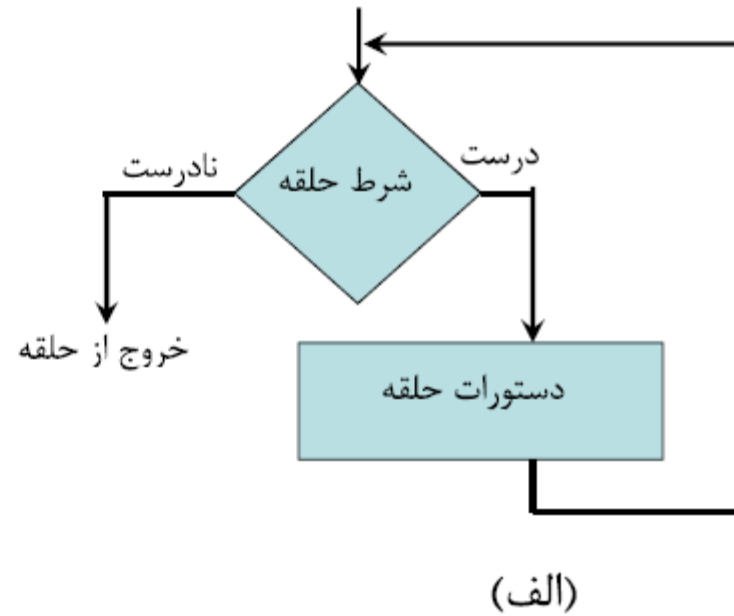
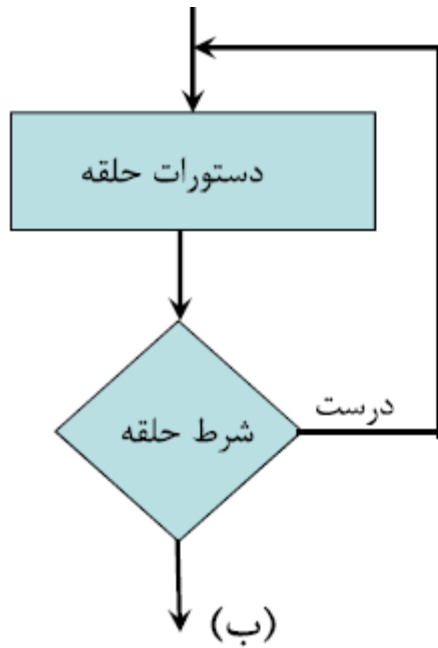


Trace

A	b
2	0
1	1
3	1

دسته بندی مسائل

- ساده (ورودی - پردازش - خروجی)
- شرطی / تصمیم گیری (ورودی - پردازش - خروجی - شرط)
- حلقه / تکرار (ورودی - پردازش - خروجی - شرط - ساختار تکرار شونده)
 - مقدار اولیه
 - شرط تکرار/خاتمه
 - گام تکرار یا شمارنده حلقه
 - عمل تکرار شونده
- ترکیبی (ساده/شرطی/حلقه)



تفاوت دو حالت: در روش ب دستورات حلقه حداقل یکبار تکرار می شوند.

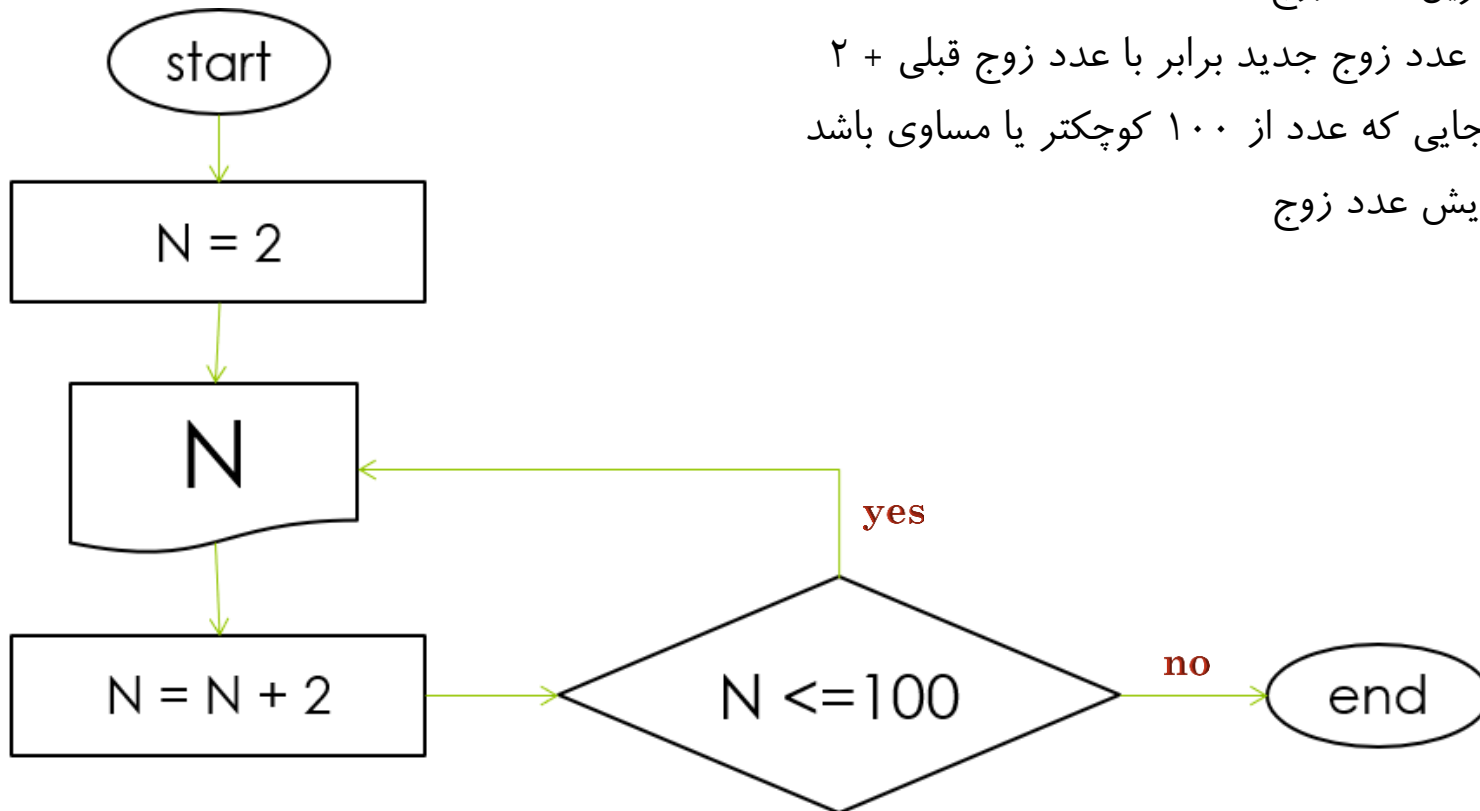
فلوچارت

روش اول

ساختار حلقه / تکرار

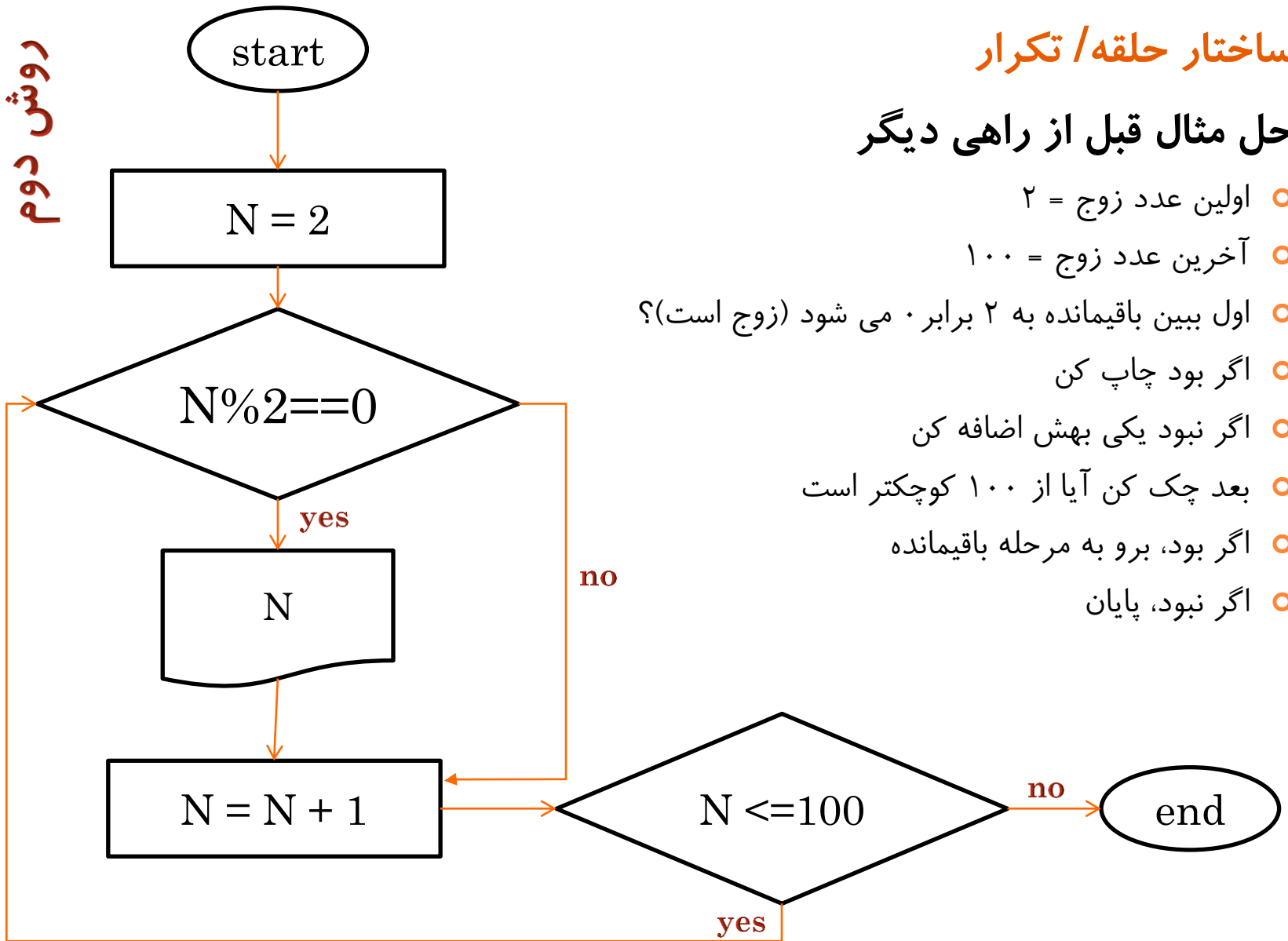
مثال : فلوچارتی بنویسید که اعداد زوج بین ۲ تا ۱۰۰ را نمایش دهد

- اولین عدد زوج = ۲
- آخرین عدد زوج = ۱۰۰
- هر عدد زوج جدید برابر با عدد زوج قبلی + ۲
- تا جایی که عدد از ۱۰۰ کوچکتر یا مساوی باشد
- نمایش عدد زوج



فلوچارت

روش دوم



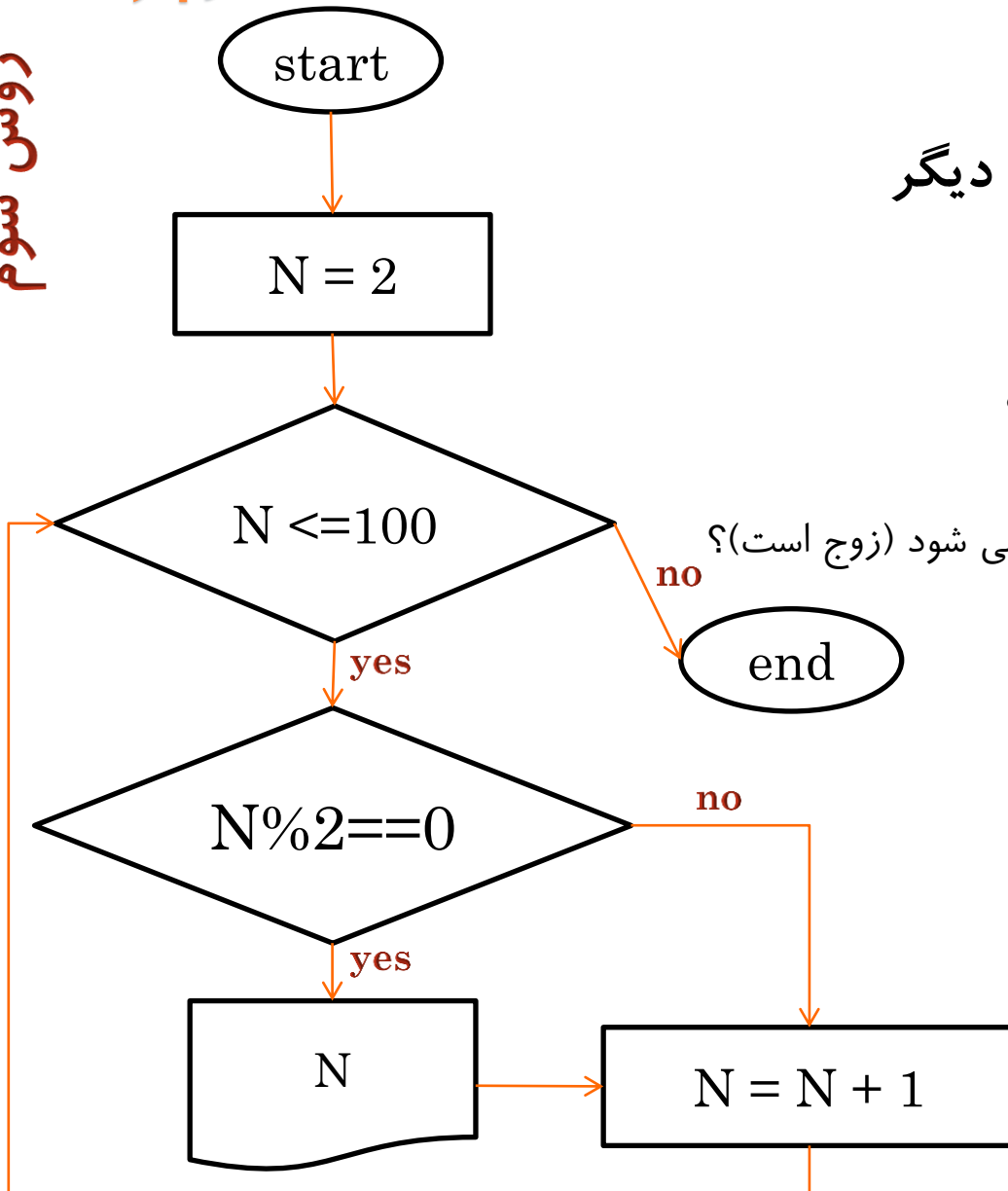
ساختار حلقه / تکرار

حل مثال قبل از راهی دیگر

- اولین عدد زوج = ۲
- آخرین عدد زوج = ۱۰۰
- اول بین باقیمانده به ۲ برابر ۰ می شود (زوج است)؟
- اگر بود چاپ کن
- اگر نبود یکی بهش اضافه کن
- بعد چک کن آیا از ۱۰۰ کوچکتر است
- اگر بود، برو به مرحله باقیمانده
- اگر نبود، پایان

فلوچارت

روش سوم



ساختار حلقه / تکرار

حل مثال قبل باز هم از راهی دیگر

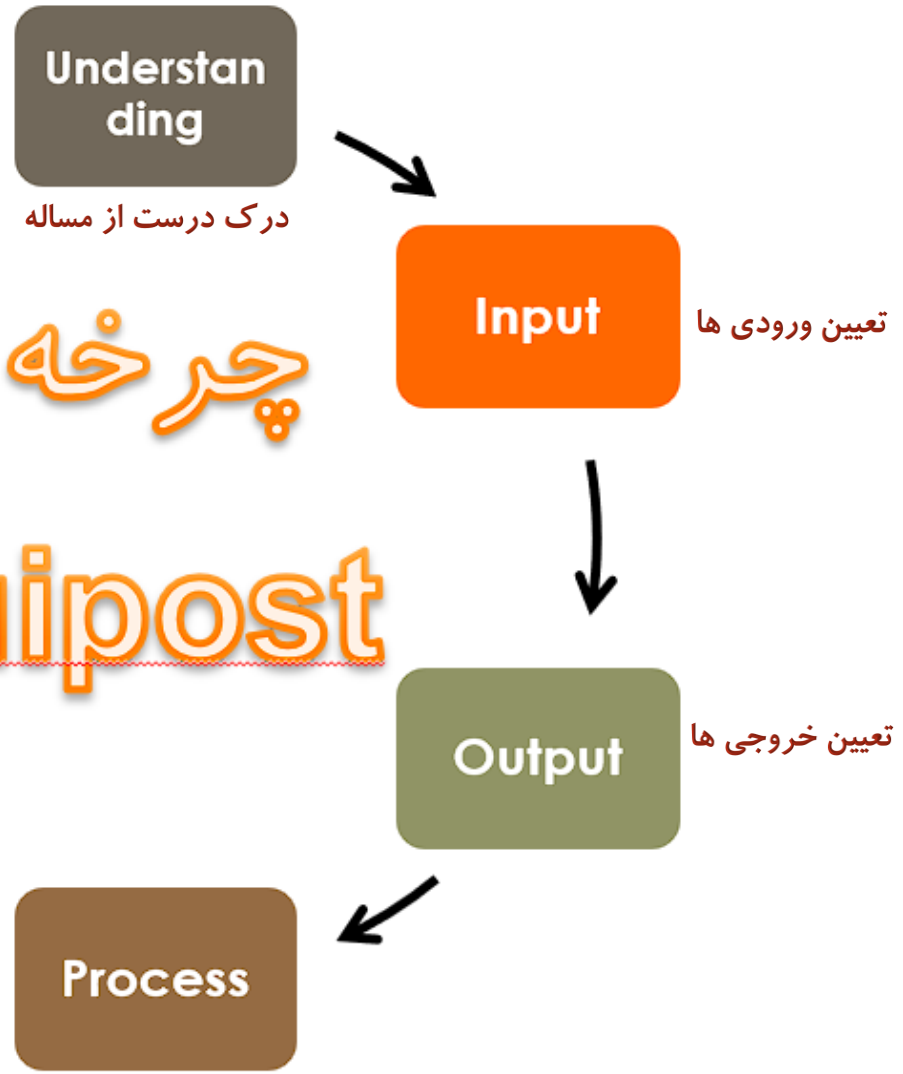
- اولین عدد زوج = ۲
- آخرین عدد زوج = ۱۰۰
- اول بین عدد کوچکتر از ۱۰۰ است
- اگر نبود پایان
- اگر بود بین باقیمانده به ۲ برابر ۰ می شود (زوج است)؟
- اگر بود چاپ کن
- بعد یکی بهش اضافه کن
- اگر نبود یکی بهش اضافه کن
- بعد برگرد به مرحله چک

چگونه یک مساله را حل کنیم

- درک درست و کامل از مسئله (تفکر سیستمی)
 - معلومات چه چیزهایی هستند؟
 - مجهولات چه چیزهایی هستند؟
- تعیین ورودی ها و خروجی ها
- تعیین پردازش و نحوه تبدیل ورودی به خروجی
- تعیین نوع ساختار راه حل (ساده، انتخابی، حلقه، ترکیبی و...)
- ترسیم فلوچارت (نوشتن راه حل)
- بررسی درستی راه حل

فلوچارت

چگونه یک مساله را حل کنیم



Understanding
درک درست از مساله

Input
تعیین ورودی ها

Output
تعیین خروجی ها

Process
تعیین پردازش و نحوه تبدیل ورودی به خروجی

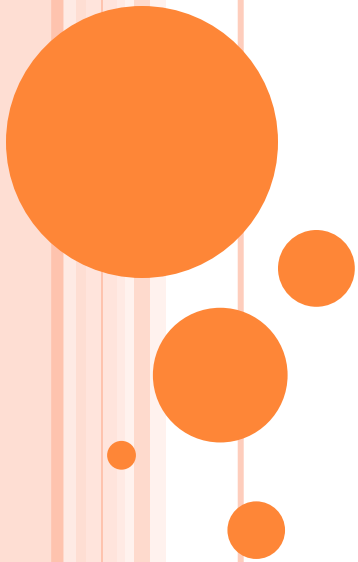
Structure
ساختار بندی و فلوچارت

Trace
اطمینان از درستی راه حل

uipost

چرخه

نمونه مسائل ساده فلوجارت



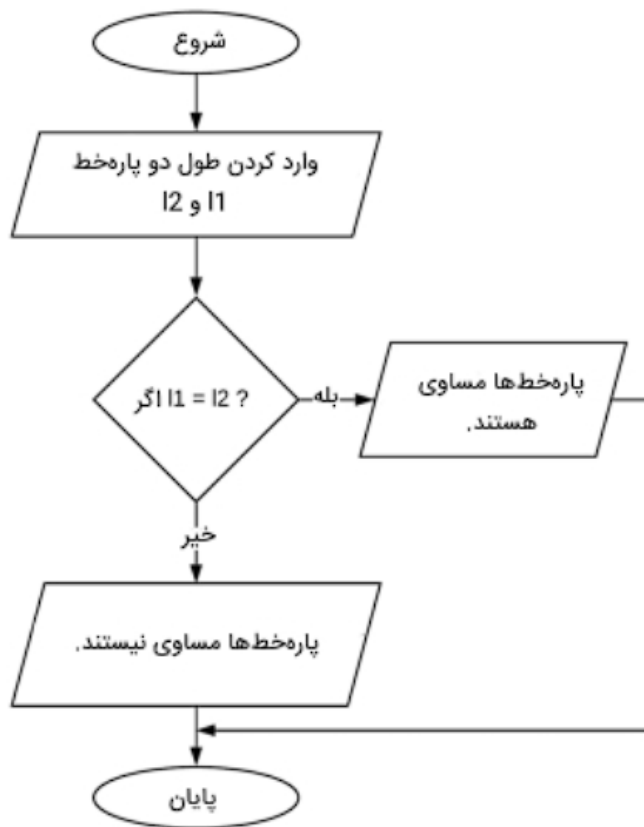
حل مساله و تمرین

فلوچارت مقایسه طول دو پاره خط و برابر بودن یا نبودن آن ها

مرحله اول: شروع

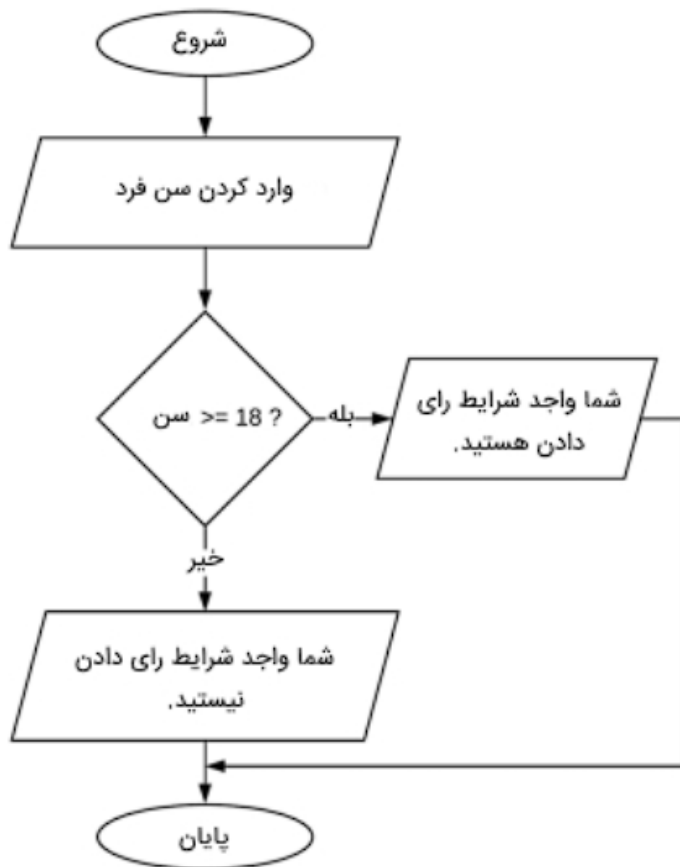
مرحله دوم: طول دو پاره خط با نام های $l1$ و $l2$ دریافت و پذیرفته می شوند.مرحله سوم: اگر $l1$ و $l2$ برابر باشند، سپس عبارت «دو پاره خط مساوی هستند.» در خروجی نشان داده می شود.مرحله چهارم: اگر $l1$ و $l2$ برابر نباشند، سپس عبارت «دو پاره خط مساوی نیستند.» در خروجی نشان داده می شود.

مرحله پنجم: پایان



حل مساله و تمرین

بررسی سن افراد جهت واجد شرایط بودن یا نبودن آنها برای رای دادن چگونه است؟



مرحله اول: شروع

مرحله دوم: سن افراد توسط برنامه دریافت می‌شود.

مرحله سوم: اگر سن فرد مساوی یا بزرگتر از عدد ۱۸ بود، سپس در خروجی عبارت «شما واجد شرایط رای دادن هستید.» چاپ می‌شود.

مرحله چهارم: اگر سن فرد کوچکتر از عدد ۱۸ بود، آنگاه در خروجی عبارت «شما واجد شرایط رای دادن نیستید.» چاپ می‌شود.

مرحله پنجم: پایان

حل مساله و تمرین

آیا با ۳ عدد می توان یک مثلث قائم الزاویه ساخت؟



جهت بررسی این موضوع که آیا سه عدد می توانند یک مثلث قائم الزاویه بسازند،

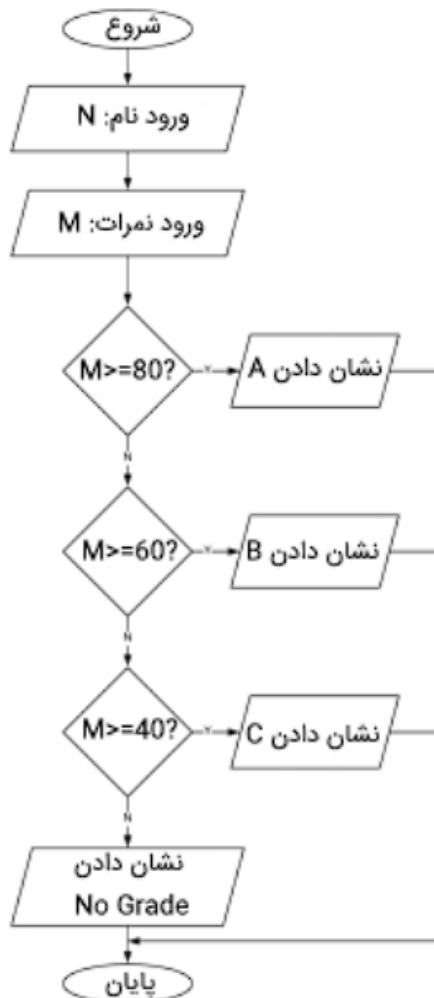
نیاز است رابطه فیثاغورس برای این ۳ عدد مورد بررسی قرار بگیرد. اگر h طول وتر مثلث قائم الزاویه باشد و p و b طول دو ضلع دیگر آن، فرمول قضیه فیثاغورس به صورت زیر نوشته می شود:

$$h = \sqrt{p^2 + b^2}$$

مساله و حل

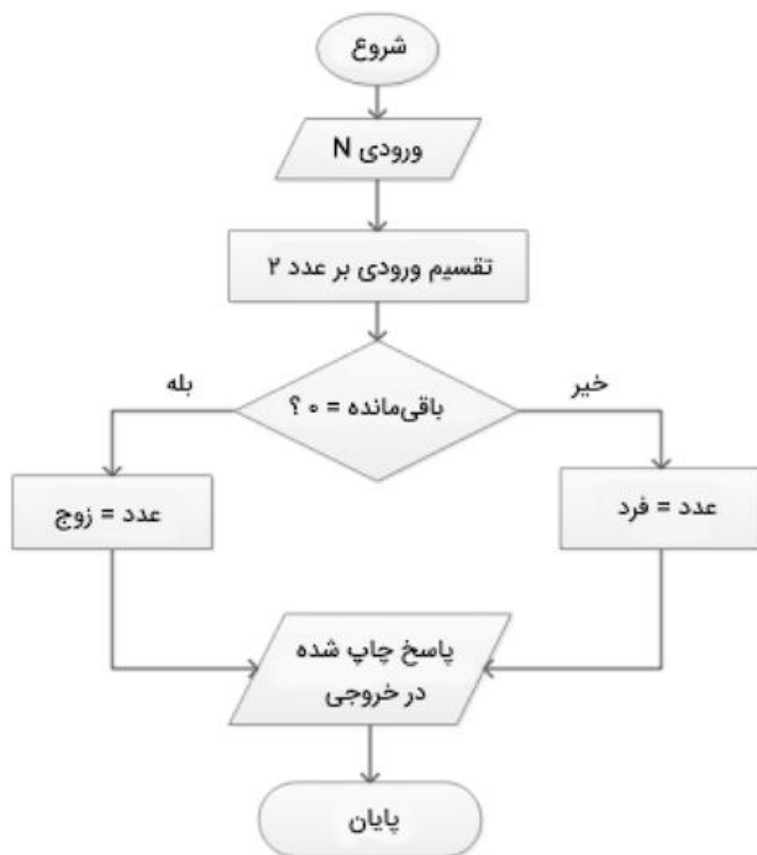
حل مساله و تمرین

دریافت نام و نمرات یک دانش آموز در پروژه به صورت جدول زیر و طراحی فلوچارت برای آن چگونه است؟



سطح نمره دانش آموز	نمرات به دست آمده دانش آموز
A	۸۰ درصد یا بیشتر
B	۶۰ درصد یا بیشتر، اما کمتر از ۸۰ درصد
C	۴۰ درصد یا بیشتر، اما کمتر از ۶۰ درصد
بدون سطح نمره (No Grade)	کمتر از ۴۰ درصد

حل مساله و تمرین

تعیین زوج یا فرد بودن عدد N و چاپ خروجی

مرحله اول: شروع

مرحله دوم: عدد N از ورودی خوانده می‌شود.

مرحله سوم: در این مرحله، عدد ورودی یعنی همان N بر عدد ۲ تقسیم می‌شود.

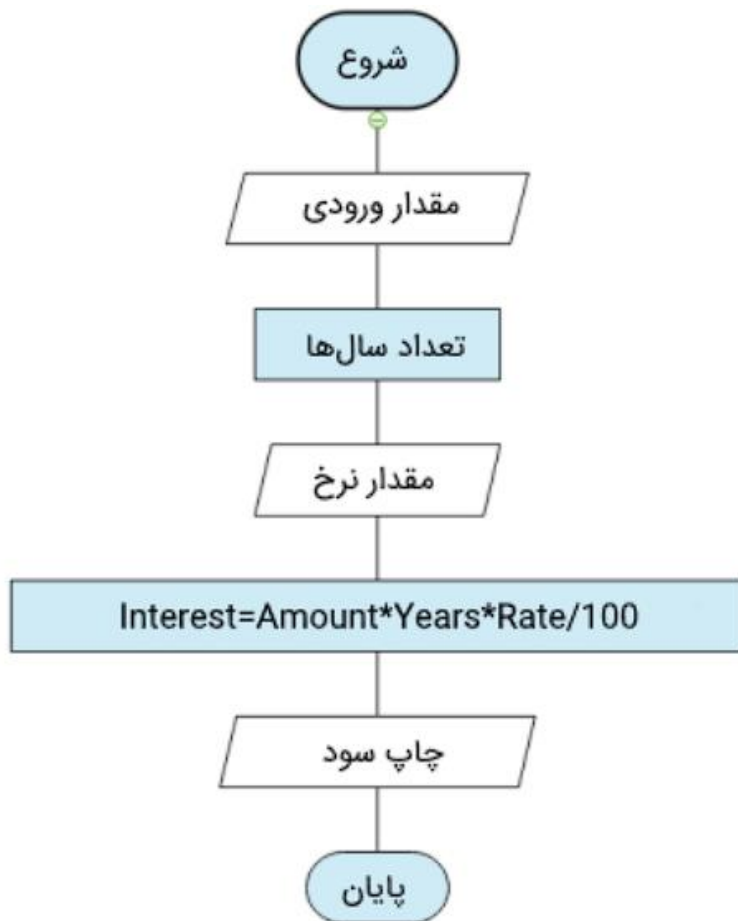
مرحله چهارم: اگر باقی مانده تقسیم عدد بر عدد ۲ برابر با صفر باشد، عدد زوج است و اگر باقی مانده این تقسیم برابر با عدد دیگری باشد، عدد ورودی N فرد است.

مرحله پنجم: چاپ خروجی

مرحله ششم: پایان

حل مساله و تمرین

فلوچارت برای محاسبه سود سپرده بانکی



مرحله اول: شروع

مرحله دوم: مقدار «رقم» (Amount) سپرده موجود در بانک خوانده می شود.

مرحله سوم: «سالهایی» (Years) خوانده می شوند که سپرده در بانک وجود داشته است.

مرحله چهارم: «نرخ» (Rate) سود بانکی خوانده می شود.

مرحله پنجم: سود بانکی با استفاده از معادله

$$Interest = \frac{Amount * Years * Rate}{100}$$

محاسبه می شود.

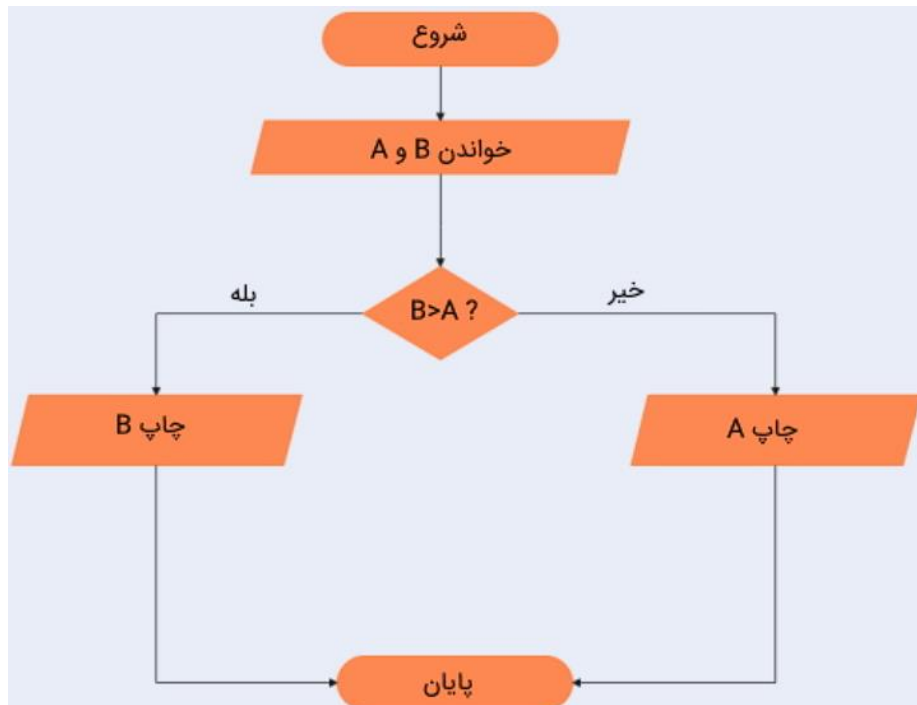
مرحله ششم: سود بانکی در خروجی چاپ می شود.

مرحله هفتم: پایان

مساله و حل

حل مساله و تمرین

فلوچارتی برای مسئله مشخص کردن بزرگترین عدد صحیح از میان دو عدد صحیح



مرحله اول: شروع

مرحله دوم: عدد صحیح A خوانده می شود.

مرحله سوم: عدد صحیح B خوانده می شود.

مرحله چهارم: اگر B بزرگتر از A باشد، سپس عدد B چاپ می شود، در غیر این صورت A چاپ خواهد شد.

مرحله پنجم: پایان

مساله و حل

حل مساله و تمرین

فلوچارتی برای محاسبه ریشه های مسئله معادله درجه دوم $ax^2 + bx + c = 0$

مرحله اول: شروع

مرحله دوم: ابتدا در این مرحله متغیرهای $a, b, c, D, x1, x2, rp, ip$ وارد برنامه می شوند.

مرحله سوم: برای محاسبه ریشه معادله درجه دوم به روش دلتا، ابتدا باید پاسخ فرمول

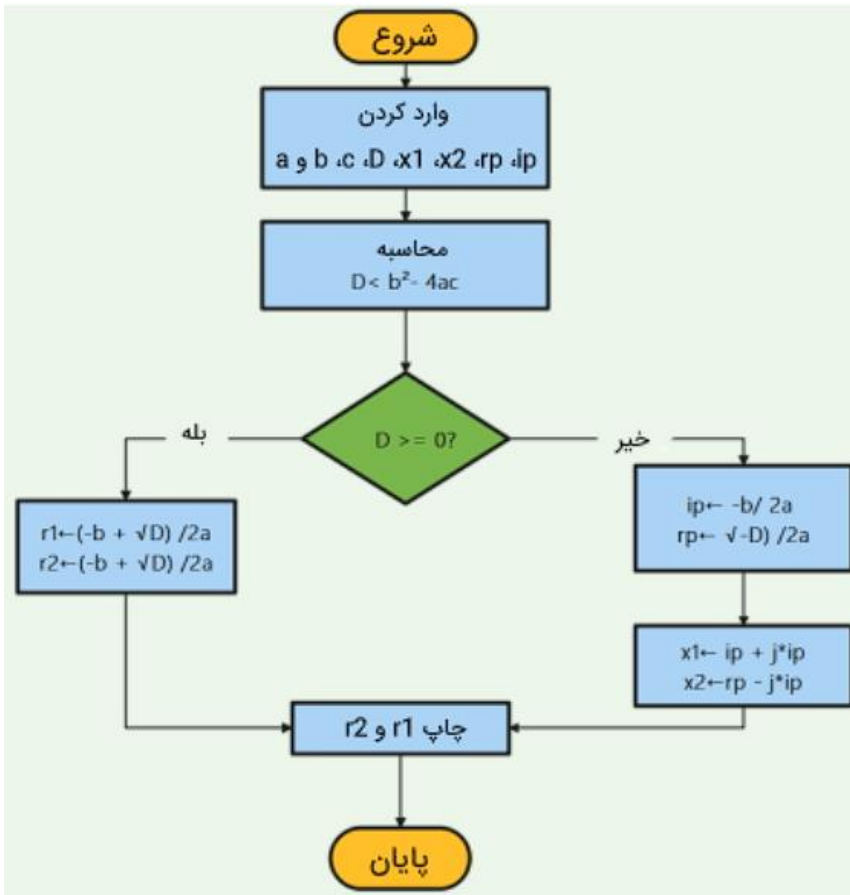
$$«D = b^2 - 4ac»$$

برای معادله درجه دوم محاسبه شود.

مرحله چهارم: اگر D بزرگتر از صفر بود یعنی معادله دارای دو ریشه است، اگر D با صفر برابر باشد یعنی معادله دو ریشه مثل هم دارد و اگر D کوچکتر از صفر باشد به این معنی است که معادله دارای دو ریشه مختلط است.

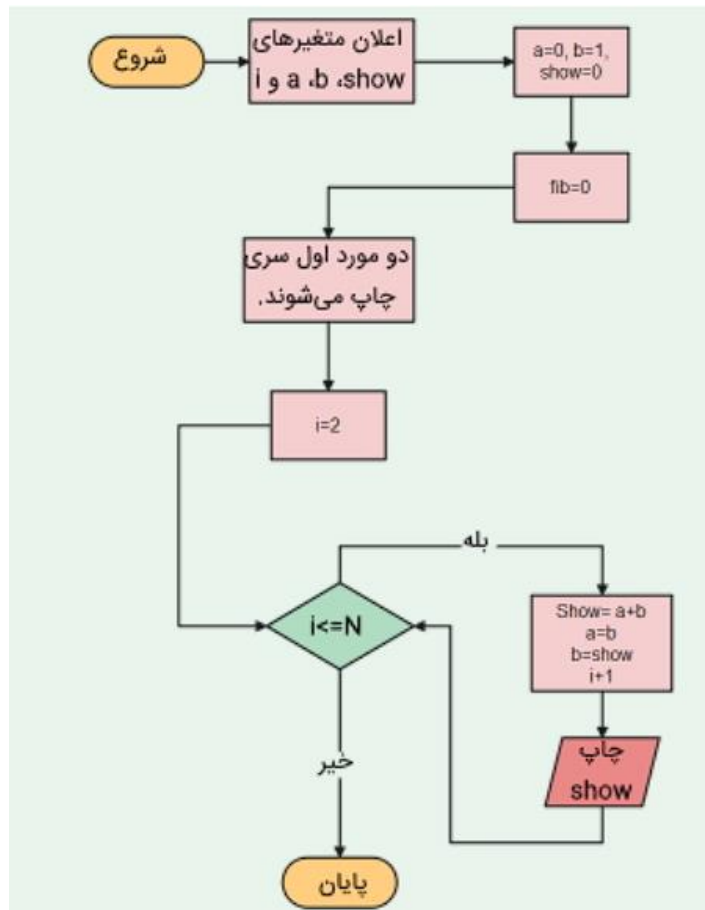
مرحله پنجم: ریشه های معادله در خروجی چاپ می شوند.

مرحله ششم: پایان



حل مساله و تمرین

فلوچارت مسئله سری فیبوناچی برای اعداد کوچکتر از ۱۰۰۰



مرحله اول: شروع

مرحله دوم: ابتدا در این مرحله متغیرهای a ، b و $show$ اعلان می شوند.

مرحله سوم: در این مرحله مقدار هر کدام از متغیرهای مرحله قبل با ارقام $a=0$ ، $b=1$ و $show=0$ مقداردهی می شوند.

مرحله چهارم: شرط سری فیبوناچی جهت حل و چاپ شدن خروجی برای مثال، مانند عدد ۱۰۰۰ وارد می شود.

مرحله پنجم: دو مورد اولیه سری فیبوناچی چاپ می شوند.

مرحله ششم: در نهایت مراحل زیر در یک حلقه پیاده سازی می شوند:

$$show = a + b \cdot 1$$

$$a = b \cdot 2$$

$$b = show \cdot 3$$

۴. در هر دور از حلقه یک عدد به مقدار متغیر a اضافه می شود.

۵. چاپ کردن متغیر $show$

مرحله هفتم: پایان

حل مساله و تمرین

فلوچارت محاسبه حاصل جمع اعداد ۱ تا ۵۰

مرحله اول: شروع

مرحله دوم: در این مرحله ابتدا دو متغیر $N = 0$ و $Sum = 0$ تعریف و اعلان می‌شوند.

مرحله سوم: سپس در این مرحله، متغیر N با استفاده از رابطه

$$N = N + 1$$

محاسبه و تعیین می‌شود.

مرحله چهارم: در این مرحله متغیر Sum به وسیله فرمول

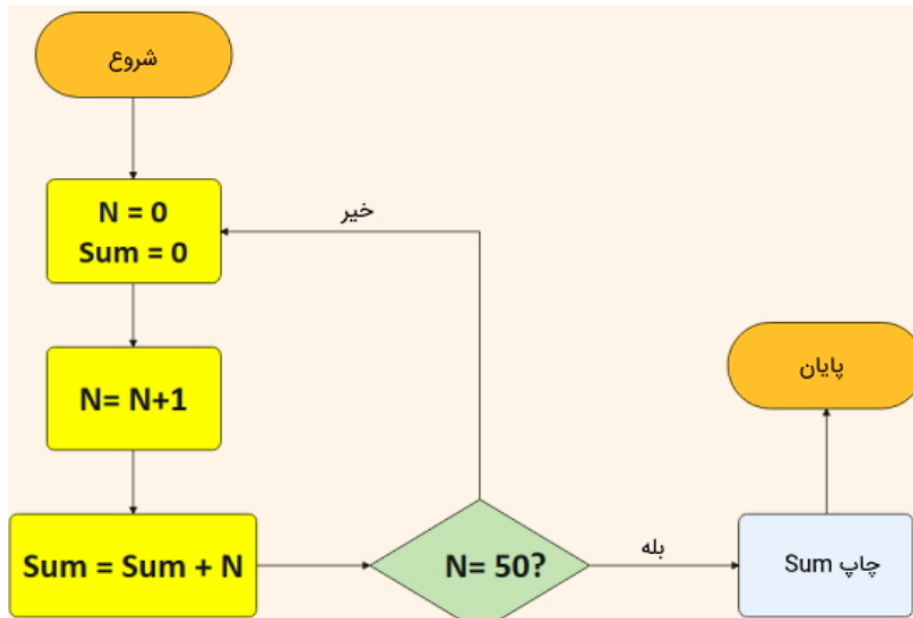
$$Sum = N + Sum$$

محاسبه شده است.

مرحله پنجم: یک حلقه بین مراحل سوم و چهارم اضافه می‌شود و تا $N = 50$ محاسبه آن ادامه دارد.

مرحله ششم: خروجی ایجاد شده در متغیر Sum چاپ می‌شود.

مرحله هفتم: پایان



مساله و حل

حل مساله و تمرین (نیاز به یادگیری آرایه ها)

فلوچارت برای یافتن بیشترین قیمت از میان ۱۰۰ عدد و کاهش ۱۰ درصدی آن

مرحله اول: شروع

مرحله دوم: ابتدا در این مرحله ۱۰۰ عدد مورد نظر سوال خوانده می‌شوند.

مرحله سوم: در این مرحله، برای حل سوال، عدد اول با عدد بعد از آن مقایسه می‌شود، آن عددی که بزرگتر است اندیس max دریافت می‌کند.

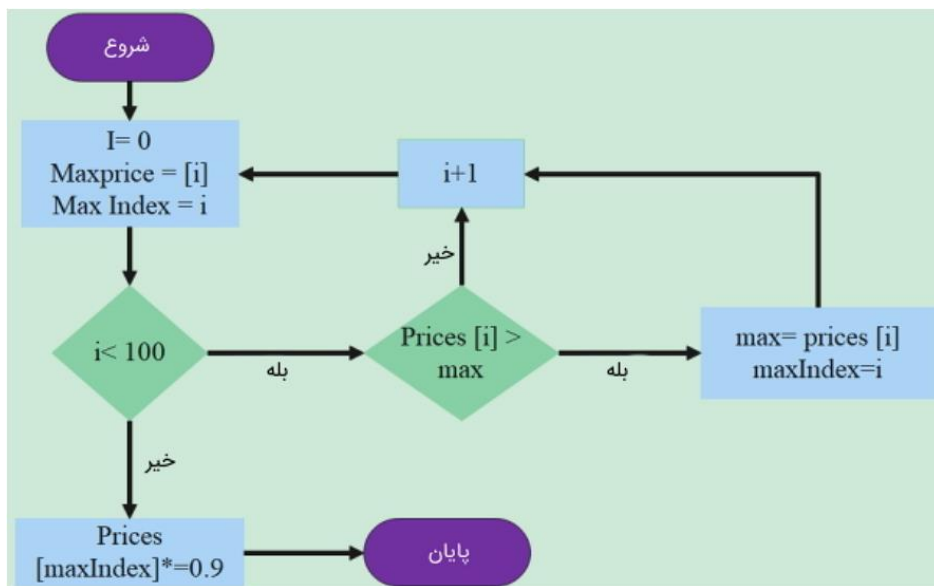
مرحله چهارم: حلقه ایجاد شده در مرحله سوم ادامه پیدا می‌کند تا بزرگترین عدد پیدا شود.

مرحله پنجم: پس از یافتن بزرگترین عدد با استفاده از مراحل قبلی، به وسیله رابط زیر مقدار ۱۰ درصد از آن کم می‌شود.

$$\text{prices}[\text{maxindex}] = \text{prices}[\text{max index}] * 0.9$$

مرحله ششم: مقدار نهایی چاپ می‌شود.

مرحله هفتم: پایان



حل مساله و تمرین

۱. فلوچارت برنامه ای را رسم کنید که مقسوم علیه های عدد N را که از کاربر دریافت می شود، نشان دهد.
(مثلا مقسوم علیه های عدد ۶ شامل ۱ و ۲ و ۳ و ۶ می باشد)
۱. فلوچارت برنامه ای را رسم کنید که ۳ عدد از کاربر دریافت کند، سپس مشخص کند آیا این سه عدد تشکیل مثلث می دهند یا نه.
(از ویژگی های مثلث این است که مجموع دو ضلع باید از ضلع سوم کوچکتر باشد)
۱. فلوچارت برنامه ای را رسم کنید که اعداد فرد بین ۱۰۰ تا ۱۰۰۰ را نمایش دهد.
(باقیمانده عدد فرد تقسیم بر ۲ برابر با ۱ می شود)
۱. فلوچارت برنامه ای را رسم کنید که ساعت کارکرد و حق الزحمه یک ساعت کارمندی را در طی یک ماه دریافت کند، سپس با احتساب ۱۰٪ مالیات و ۷٪ بیمه، حقوق وی را محاسبه نماید و آن را نشان دهد.



با تشکر از همراهی شما

محمد سعید صفایی صادق