

باسمه تعالی

برنامه نویسی کامپیوتر

تنها درس دانشگاهی رشته مهندسی مکانیک می باشد که مستقیماً به طرح اصول اولیه برنامه نویسی می پردازد.

در بسیاری از فعالیت های تحقیقاتی، پیاده سازی روش و شبیه سازی برخی فرآیندها نیازمند برنامه نویسی است.

بنابراین یادگیری دقیق این درس به همراه ارائه پروژه های عملی که لازمه این درس می باشد جزء اهم مسائل می باشد .

طرح درس

شماره هفته آموزشی	مبحث	توضیحات
۱	آشنایی با سرفصل درس، منابع درس و نحوه ارزشیابی - بیان مقدمه و مرور مختصر مبانی کامپیوتر	
۲	الگوریتم و فلوچارت - آشنایی با مفهوم الگوریتم، فرآیند حل مسئله، قواعد تبدیل الگوریتم به برنامه کامپیوتری	
۳	الگوریتم و فلوچارت - آشنایی با فلوچارت، اجزا فلوچارت، تبدیل الگوریتم به فلوچارت	
۴	الگوریتم و فلوچارت - ساختارهای شرطی، جدول ردیابی	
۵	الگوریتم و فلوچارت - حلقه های با تکرار مشخص و نامشخص، حلقه های تو در تو	
۶	الگوریتم و فلوچارت - آرایه ها، مرتب سازی، جستجو	
۷	زبان - C معرفی زبان C، آشنایی با IDE، ساختار کلی برنامه نویسی در زبان C	
۸	زبان - C انواع متغیرها، انواع داده، نحوه نامگذاری، مقدار دهی اولیه، توابع ورودی-خروجی	
۹	زبان - C ادامه توابع ورودی-خروجی Printf و Scanf، طول میدان، توابع دریافت کاراکتر، کد اسکی	
۱۰	زبان - C انواع عبارات (ریاضی، شرطی، منطقی)	
۱۱	زبان - C ساختارهای کنترل اجرای برنامه، دستورات شرطی (if, if-else, if-else if, switch-case)	
۱۲	زبان - C حلقه ها (for, while, do-while)	
۱۳	زبان - C نحوه ایجاد تابع در زبان C، نمونه اولیه تابع، متغیرهای محلی، بلاکی و کلی، توابع بازگشتی، فراخوانی مرجع	
۱۴	زبان - C آرایه ها، آرایه های یک بعدی، مقدار دهی اولیه، توابع و آرایه ها، مرتب سازی و جستجو، آرایه های دوبعدی	
۱۵	زبان - C رشته ها، مقدار دهی اولیه، توابع کار با رشته ها، ساختار ها، مقدار دهی اولیه، ارسال به توابع	
۱۶	زبان - C فایل، دستورات ایجاد فایل، باز کردن فایل، انتقال داده به فایل، خواندن داده از فایل	

منابع درس

منابع و مآخذ درس	فایل های PDF درس برنامه نویسی جامع به زبان C، مونس علی فرمنش، انتشارات جهاد دانشگاهی دانشگاه صنعتی امیر کبیر، ۱۳۷۸ برنامه نویسی به زبان C، عین اله جعفر نژاد قمی، انتشارات جهاد دانشگاهی مشهد، ۱۳۷۱
------------------	--

شیوه ارزشیابی

ارزشیابی درس	تمرین (فلوچارت، کدنویسی در سایت کوئرا) کوئیز حضور و غیاب (نمره تشویقی) ۴ نمره ۲ نمره هر جلسه ۰,۰۴ امتحان میان ترم اول – چهار گزینه ای (از <i>choo</i> تا <i>cho5</i>) ۵ نمره امتحان میان ترم دوم – کدنویسی (کوئرا) ۵ نمره امتحان پایان ترم – چهار گزینه ای (از <i>cho1</i> تا <i>cho11</i>) ۵ نمره
--------------	---

هدفهای کلی

شناخت کامپیوترهای نسل قدیم و امروزی
شناخت سخت افزارهای لازم برای کامپیوترهای شخصی
بررسی نرم افزارها و انواع آن

❖ کامپیوترهای دیجیتال – نسل اول تا چهارم

نسل اول

اولین کامپیوتر در سال ۱۹۴۴ در دانشگاه هاروارد و نوع کاملتر آن در سال ۱۹۴۶ در دانشگاه پنسیلوانیا به نام انیاک، برای حل مسایل مربوط به انفجار، جهت ارتش آمریکا توسط دکتر ساخته و تکمیل شد. در این ماشین ۱۹۰۰۰ لامپ خلا استفاده شده بود و برای انرژی مصرفی لامپ ها و همچنین دستگاههای تهویه و خنک کننده ماشین حدود ۱۳۰ کیلو وات انرژی الکتریکی مصرف می شد. سطحی را معادل ۹۰۱۵ متر مربع اشغال می کرد، ولی سرعت زیادی داشت و ۵۰۰۰ جمع و ۳۵۰ ضرب را در ۱ ثانیه به انجام می رسانید. در سال ۱۹۵۲ اولین کامپیوتری که قادر به ذخیره کردن برنامه بود به نام ادواک، توسط دکتر نیومن، ساخته شد که اساس کامپیوترهای امروزی قرار گرفت. در سال ۱۹۴۸ کامپیوتر دیگری توسط شرکت IBM ساخته شد که سرعت عمل زیادی داشت.

تا قبل از سال ۱۹۵۵ برای فعالیتهای تجاری یا کارهای علمی، کامپیوترهای ویژه ای ساخته می شد که به کامپیوترهای نسل اول معروفند.

مشخصات کلی کامپیوتر های نسل اول:

۱. سرعت عمل آنها حدود یک هزارم ثانیه بود.
۲. حافظه آنها دارای ظرفیت ۲۰۰۰ تا ۴۰۰۰ کلمه بود.
۳. دارای کاربردهای ویژه تک منظوره بودند.
۴. کلیه برنامه ها به زبان ماشین، نوشته می شد.
۵. در آنها لامپ خلا و رله به عنوان حافظه استفاده می شد.

❖ کامپیوترهای دیجیتال – نسل اول تا چهارم

نسل دوم

در اوایل دهه ۱۹۵۰ با ورود ترانزیستور به بازار و استفاده از آن در کامپیوتر و همچنین به کار بردن حلقه های کوچک مغناطیسی به عنوان حافظه ، تغییرات عمده ای در کامپیوتر ها ایجاد گردید. اختراع ترانزیستور ، کامپیوتر های جدید را کوچکتر ، سبکتر و قابل اعتمادتر کرد و همچنین مصرف برق آنها را به مقدار زیاد کاهش داد. کاربرد حلقه های کوچک مغناطیسی به عنوان حافظه نیز، سرعت فراوانی به کامپیوترها بخشید.

اولین کامپیوتری که در ایران نصب گردید از نسل دوم و مدل IBM 1620 بود که در سال ۱۳۴۱ در کنسرسیوم نفت تهران به کار گرفته شد و همچنین سرشماری سال ۱۳۴۵ نیز با استفاده از کامپیوترهای نسل دوم انجام گردید.

مشخصات کلی کامپیوتر های نسل دوم:

۱. از ترانزیستور در آنها استفاده شد.
۲. سرعت عمل آنها حدود یک میلیونیم ثانیه بود.
۳. ظرفیت حافظه آنها حدود ۳۰۰۰۰ کلمه بود و حافظه های کمکی نیز در این نسل به وجود آمدند.
۴. دارای کاربردهای عمومی یا همه منظوره بودند.
۵. زبانهای برنامه نویسی آنها ، فوق العاده آسان بود.
۶. دارای حجم بسیار کمتری بودند.
۷. از حلقه های کوچک مغناطیسی به عنوان حافظه در آنها ، استفاده می شد.

❖ کامپیوترهای دیجیتال – نسل اول تا چهارم

نسل سوم

برای ساختن کامپیوترهای سریعتر و قویتر کوششها همچنان ادامه داشت تا در اوایل ۱۹۶۰ اولین کامپیوتر نسل سوم به بازار عرضه شد. جدیدترین تحول در تکامل کامپیوترها، ساختن وسایل ضبط اطلاعات با قابلیت دسترسی مستقیم در این نسل بود. به این ترتیب کاربران توانستند به هر یک از اجزا اطلاعات ذخیره شده در یک مجموعه عظیم اطلاعاتی، در کسری از ثانیه دسترسی پیدا کنند. علاوه بر آن در این نسل از کامپیوترها، سعی شده که قطعات مدارها را هرچه کوچکتر و با حجم کمتر بسازند و بدین ترتیب مدارهای مجتمع به وجود آمدند.

مشخصات کلی کامپیوتر های نسل سوم:

۱. پیشرفت های سخت افزاری
(الف) مینیاتوری کردن (تقلیل حجم دستگاهها و اجزای آنها) (ب) افزایش ظرفیت حافظه به چندین برابر قبل
(ج) استفاده از دستگاه های واسطه، با قابلیت دسترسی مستقیم (د) قدرت ارتباط با نقاط دور و متعدد
۲. پیشرفت های نرم افزاری
(الف) هماهنگی بیشتر با سخت افزار (ب) هماهنگی بیشتر با سیستم عامل
(ج) پیشرفت در زبانهای برنامه نویسی و به کارگیری زبان های سطح بالا
۳. عملیات و بهره برداری
(الف) استفاده از روش های پردازش آنلاین (ب) اجرای همزمان چند برنامه با یکدیگر

❖ کامپیوترهای دیجیتال – نسل اول تا چهارم

نسل چهارم

تقسیم بندی و تفکیک نسل های کامپیوتری تا قبل از نسل چهارم، به لحاظ تغییرات عمده در پیشرفت و تکامل کامپیوتر در هر نسل، به سهولت صورت گرفت. در اوایل سال ۱۹۷۰ تکنیکهای جدیدتری در ساخت و بهره گیری از کامپیوترها به کار برده شد که بسیاری از دست اندرکاران آن را نسل چهارم نامیدند. مهمترین تغییرات در سخت افزار کامپیوترهای نسل چهارم، به کارگرفتن مدارهای مجتمع با تراکم زیاد و تراکم خیلی زیاد است. در نسل سوم از تراکم کم و تراکم متوسط بهره گرفتند ولی در نسل چهارم از تراکم بالا، خیلی بالا و فوق العاده بالا بهره می گیرند. نسل چهارم همچنین از حافظه نیمه هادی و میکرو پروسسور و بهره می گیرد.

توسعه و پیشرفت سخت افزار کامپیوترهای فعلی، در مقایسه با نسل های قبلی با بررسی چند عامل نظیر سرعت، اندازه، هزینه و ظرفیت حافظه روشن می گردد.

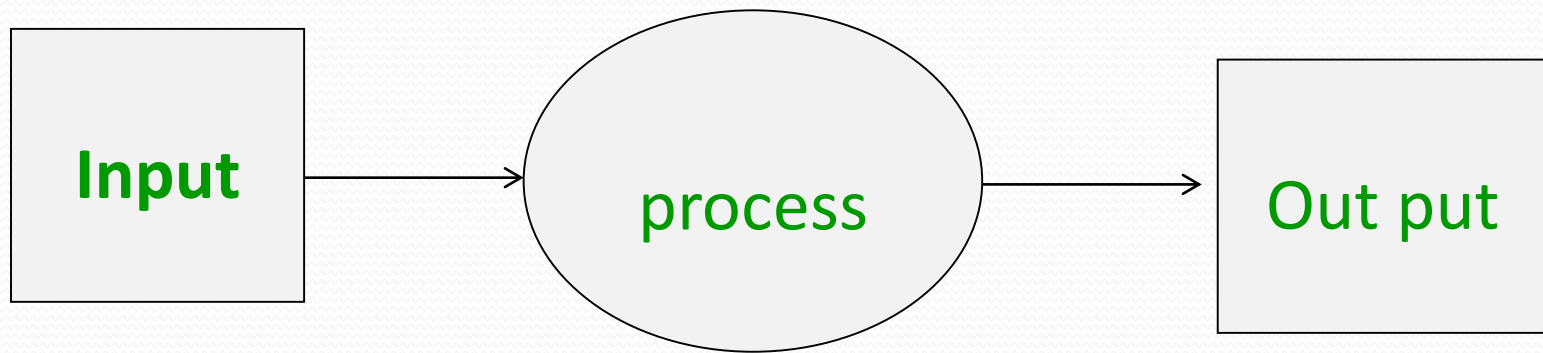
❖ کامپیوترهای امروزی

کامپیوترهای امروزی با بکارگیری ریزپردازنده به کامپیوترهای نسل چهارم معروفند. البته نسل‌های جدید دیگر کامپیوترها نیز به بازار ارائه می‌شود. در کامپیوترهای امروزی سرعت پردازش بسیار بالا، حجم اجزاء سخت‌افزاری بسیار کوچک، حجم حافظه بالا و غیره آنها را از نسل‌های دیگر متمایز می‌سازد.

❖ اجزاء تشکیل دهنده کامپیوتر

- سخت افزار

- نرم افزار



کامپیوترهای امروزی معمولاً از قطعات زیر تشکیل می شوند:

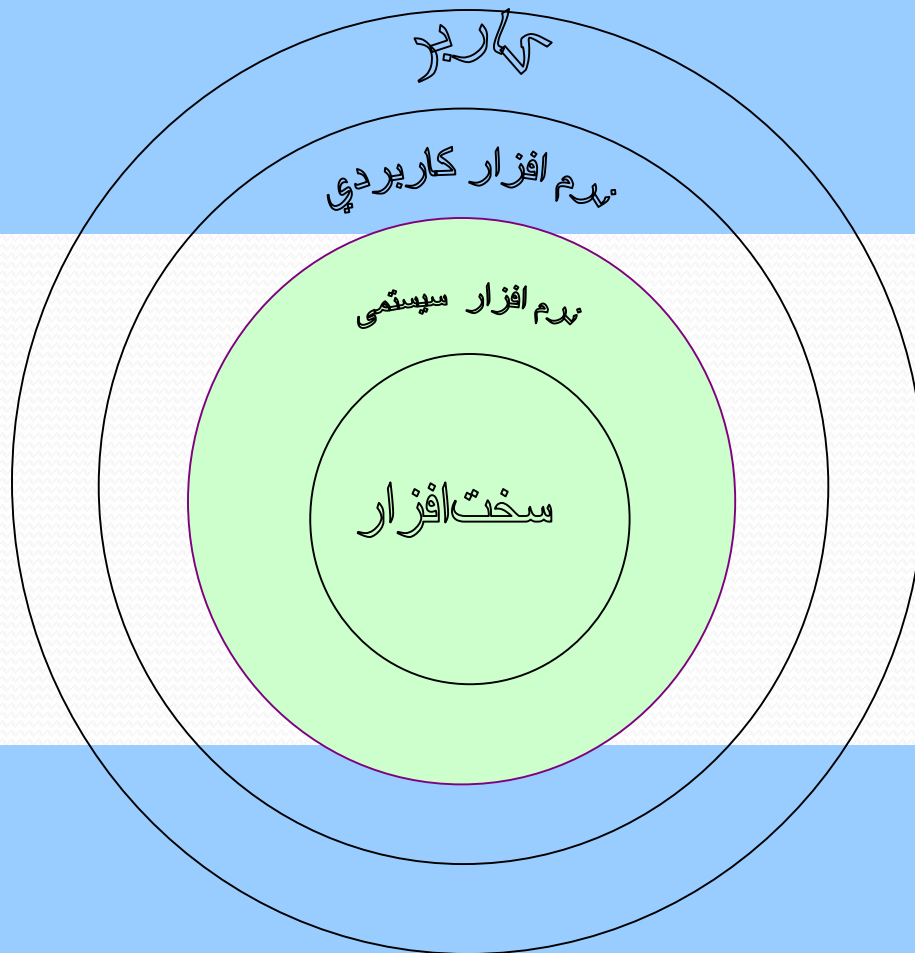
- دستگاههای ورودی
- حافظه های جانبی
- حافظه های اصلی
- واحد پردازشگر مرکزی
- دستگاههای خروجی

❖ نرم افزار

نرم افزار یکی از بخش های اساسی کامپیوتر به شمار می آید، که در واقع سخت افزار را بکار می گیرد. عبارت دیگر رابط بین کاربر و سخت افزار را نرم افزار می نامند. نرم افزار در حقیقت روح و جان یک کامپیوتر است، که به سخت افزار هویت می بخشد.

نرم افزار سیستم عامل

- سیستم عامل (OS: Operating System) مشهورترین نوع نرم افزارهای سیستمی می باشد. که مدیریت منابع سیستمی را بر عهده دارد. سیستم عامل، همچنین ارتباط بین کاربر و اجزاء سخت افزاری و نرم افزاری دیگر را برقرار می کند.



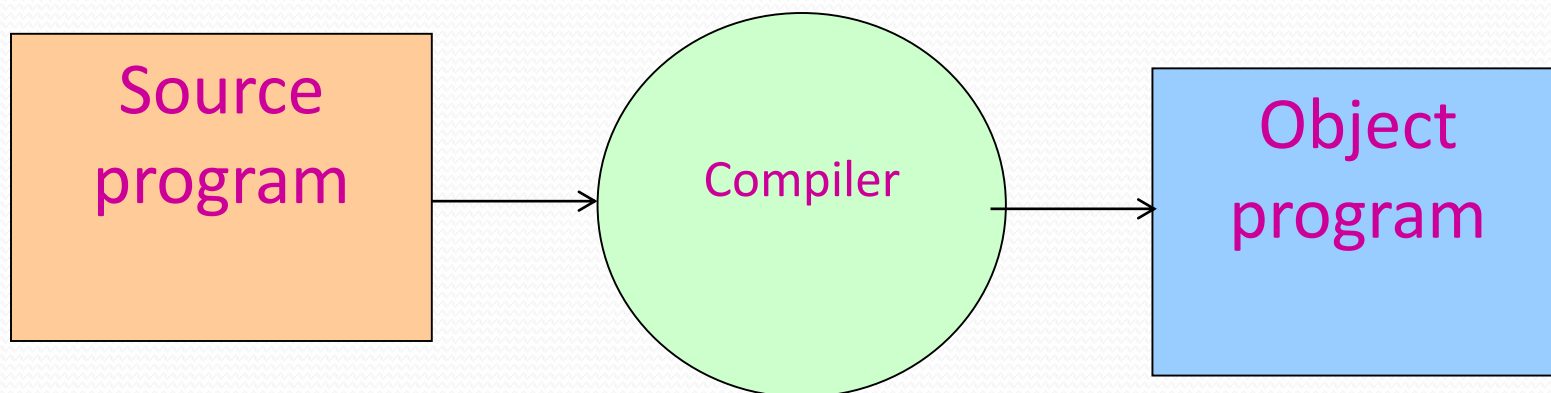
❖ زبانهای برنامه‌نویسی

- نرم‌افزارها توسط زبانهای برنامه‌نویسی نوشته می‌شوند. زبانهای برنامه‌نویسی، یک سیستم ارتباطی هستند که توسط آنها می‌توان دستورات لازم را به ماشین انتقال داد.
- هر زبان برنامه‌نویسی به مجموعه‌ای از علایم، قواعد و دستورالعمل‌ها گفته می‌شود که امکان ارتباط با کامپیوتر را جهت بیان کاری یا حل مسئله‌ای فراهم می‌کند.

در حالت کلی زبانهای برنامه نویسی را به سه دسته زیر تقسیم بندی می کنند:

- زبانهای سطح بالا
- زبانهای سطح پایین
- زبانهای سطح میانی

کامپایلر برنامه نوشته در یک زبان سطح بالا را به برنامه مقصد تبدیل می کند.



❖ الگوریتم و فلوچارت

هدفهای کلی

شناخت حل مسئله و ارائه الگوریتم
شناخت اجزاء لازم برای حل مسئله
بررسی صحت الگوریتم

مقدمه

در زندگی روزمره، انسان با مسائل مختلفی روبروست و برای هر کدام از این مسائل (حل مشکلات) راه حلی و روشی را برمیگزیند. 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.

❖ تعریف الگوریتم

الگوریتم مجموعه‌ای از دستورالعمل‌ها، برای حل مسئله می‌باشد که شرایط زیر را باید دارا باشد:

دقیق باشد.

جزئیات کامل حل مسئله را داشته باشد.

پایان پذیر باشد.

❖ مراحل الگوریتم

برای حل یک مسئله باید الگوریتم آن مسئله را مشخص کنیم (یا بیابیم).
که اصطلاحاً طراحی الگوریتم برای آن مسئله نامیده می‌شود. در طراحی
الگوریتم معمولاً سه مرحله زیر را از هم جدا می‌کنند:

- خواندن داده‌ها
- انجام محاسبات
- خروجی‌ها

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

ورودیها

a , b

انجام محاسبات

جمع دو عدد

خروجیها

مجموع دو عدد

۰- شروع

۱- a , b را بخوان.

۲- مجموع a , b را محاسبه و در sum قرار بده.

۳- sum را در خروجی چاپ کن

۴- پایان

مثال: الگوریتمی بنویسید که سه عدد از ورودی دریافت کرده مجموع و میانگین سه عدد را محاسبه و چاپ کند.

ورودیها

b,c,a

انجام محاسبات

محاسبه مجموع
محاسبه میانگین

خروجی ها

چاپ مجموع
چاپ میانگین

0- شروع

1- سه عدد از ورودی بخوان

2- مجموع سه عدد را محاسبه و در sum قرار بده.

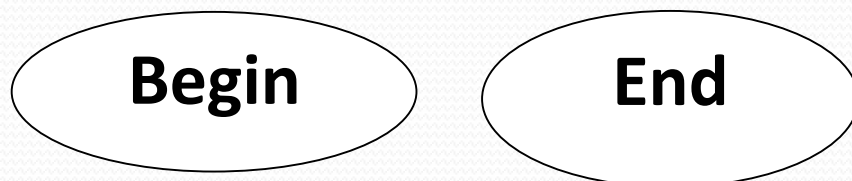
3- sum را بر سه تقسیم کرده، در ave قرار بده.

4- sum , ave را در خروجی چاپ کن.

5- پایان.

معمولا درک یک الگوریتم با شکل راحتتر از نوشتن آن بصورت متن می باشد. لذا الگوریتم را با فلوچارت (flowchart) نمایش می دهند. فلوچارت از شکل های زیر تشکیل می شود.

علامت های شروع و پایان: که معمولا از یک بیضی استفاده می کنند:



علامت های ورودی و خروجی: که معمولا از متوازی الاضلاع استفاده می شود:

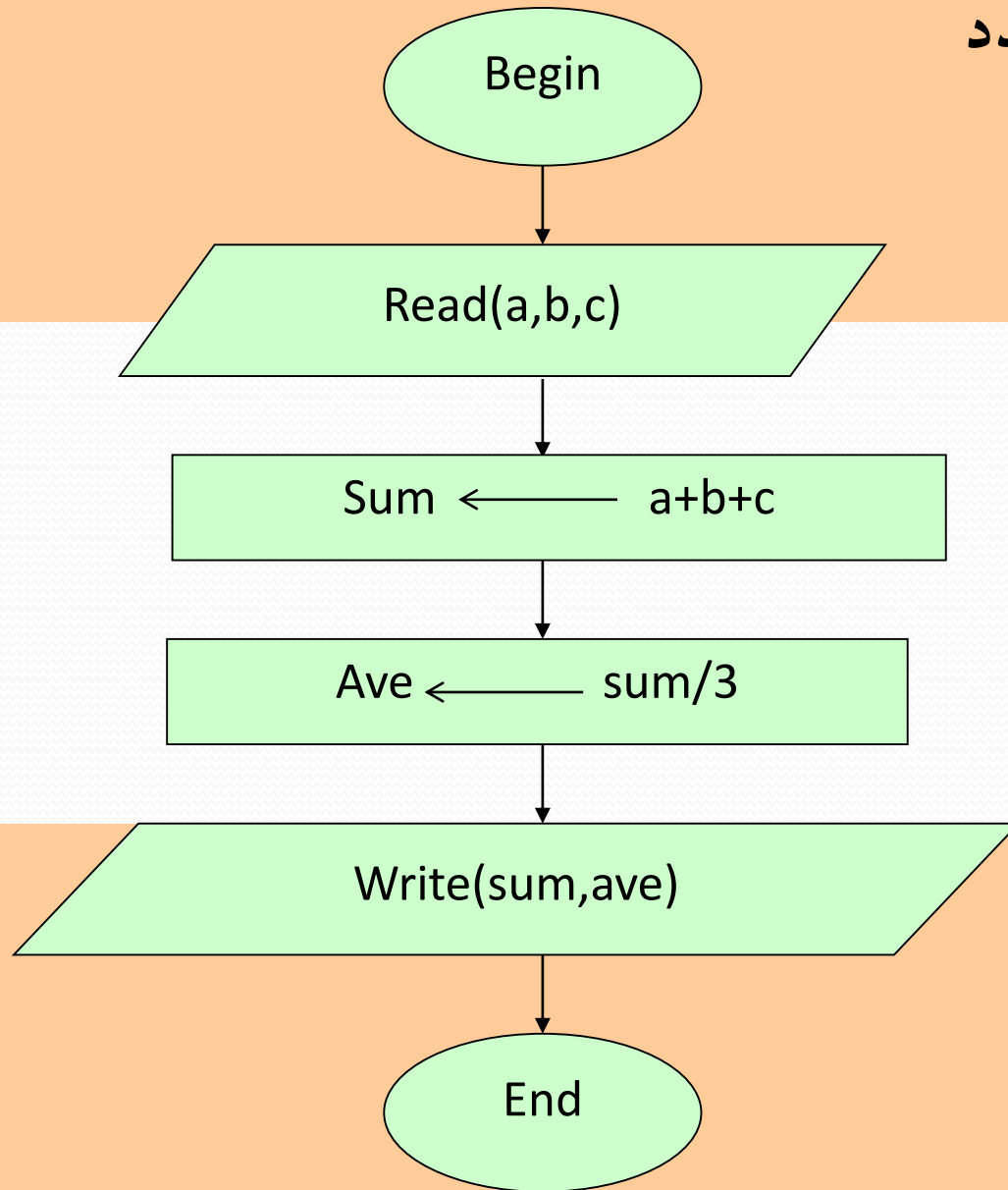


جایگزین یا محاسبات

● علامتهای محاسباتی و جایگزینی: برای نمایش دستورات جایگزینی و محاسباتی از مستطیل استفاده می کنند:

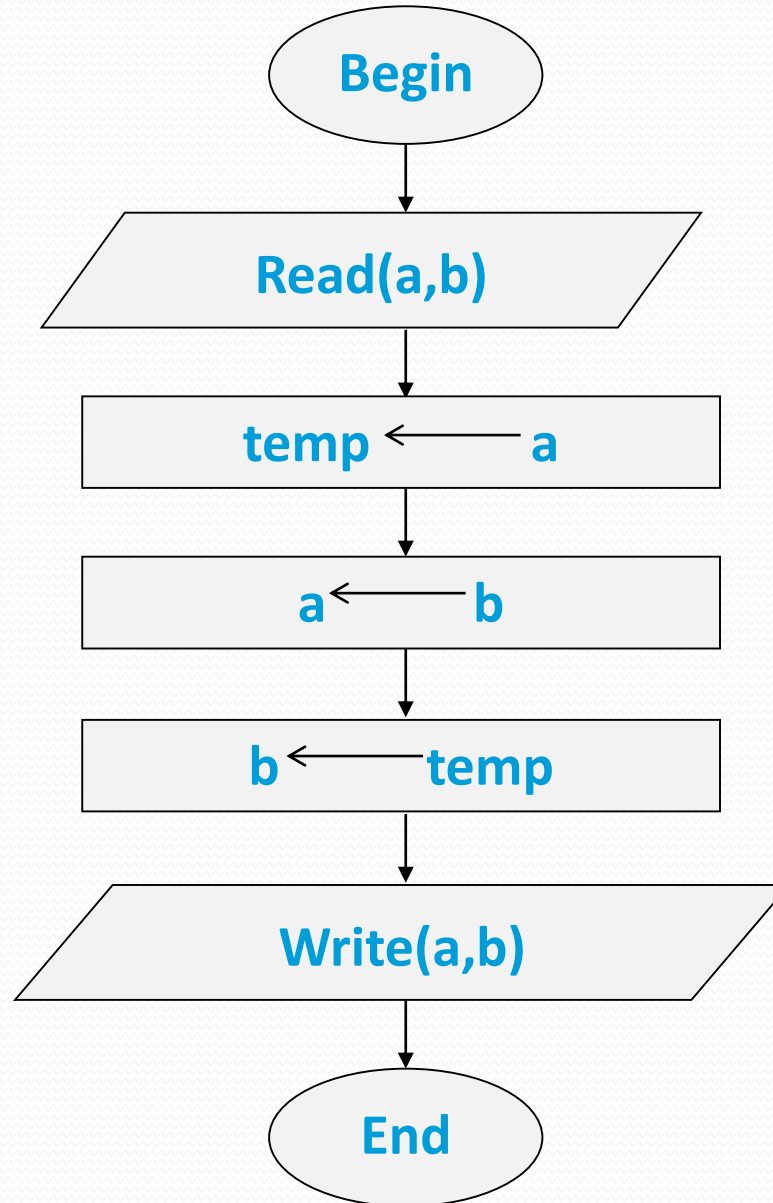
- علامت شرط: برای نمایش شرط از لوزی استفاده می شود.
- علامت اتصال: برای اتصال شکل های مختلف بهم از فلش های جهت دار استفاده می کنند.

فلوچارت مجموع سه عدد



مثال: فلوچارتی رسم نمائید که دو عدد از ورودی دریافت کرده سپس محتویات دو عدد را با هم جابجا نماید.

برای حل این مسئله a , b را دو متغیر که در آنها دو عدد خوانده شده، قرار می‌گیرند در نظر می‌گیریم. سپس با استفاده از یک متغیر کمکی محتویات این دو عدد را جابجا می‌کنیم:



$A=10$

$B=5$

$A=A-B=5$

$B=B+A=10$

$A=B-A=5$

تمرین سری اول

۱- فلوچارتی رسم نمائید که طول و عرض مستطیل را از ورودی دریافت کرده محیط و مساحت آنرا محاسبه و چاپ کند.

۲- فلوچارتی رسم نمائید که شعاع دایره‌ای را از ورودی دریافت کرده، محیط و مساحت آنرا محاسبه و چاپ نماید.

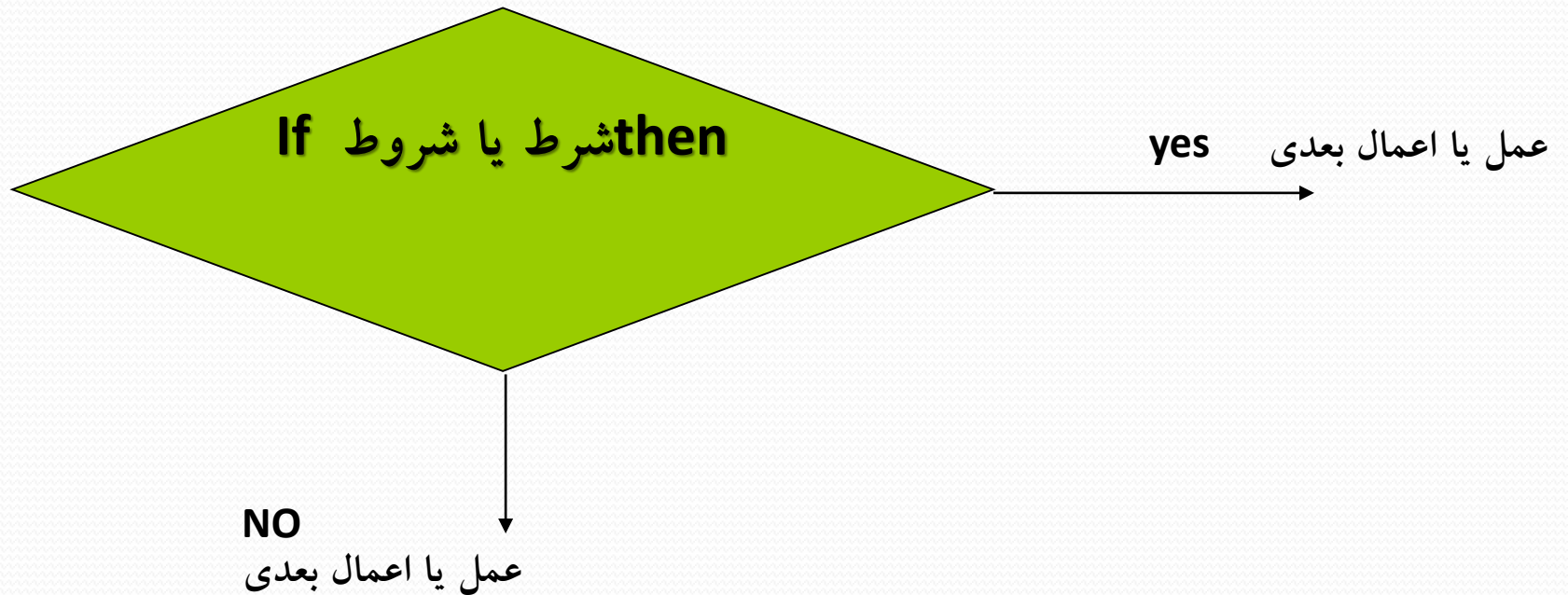
۳- فلوچارتی رسم کنید که سه عدد **Third , second, first** را از ورودی دریافت کرده، محتویات آنها را جابجا نموده، حاصل را در خروجی چاپ کند.

❖ دستورالعمل‌های شرطی

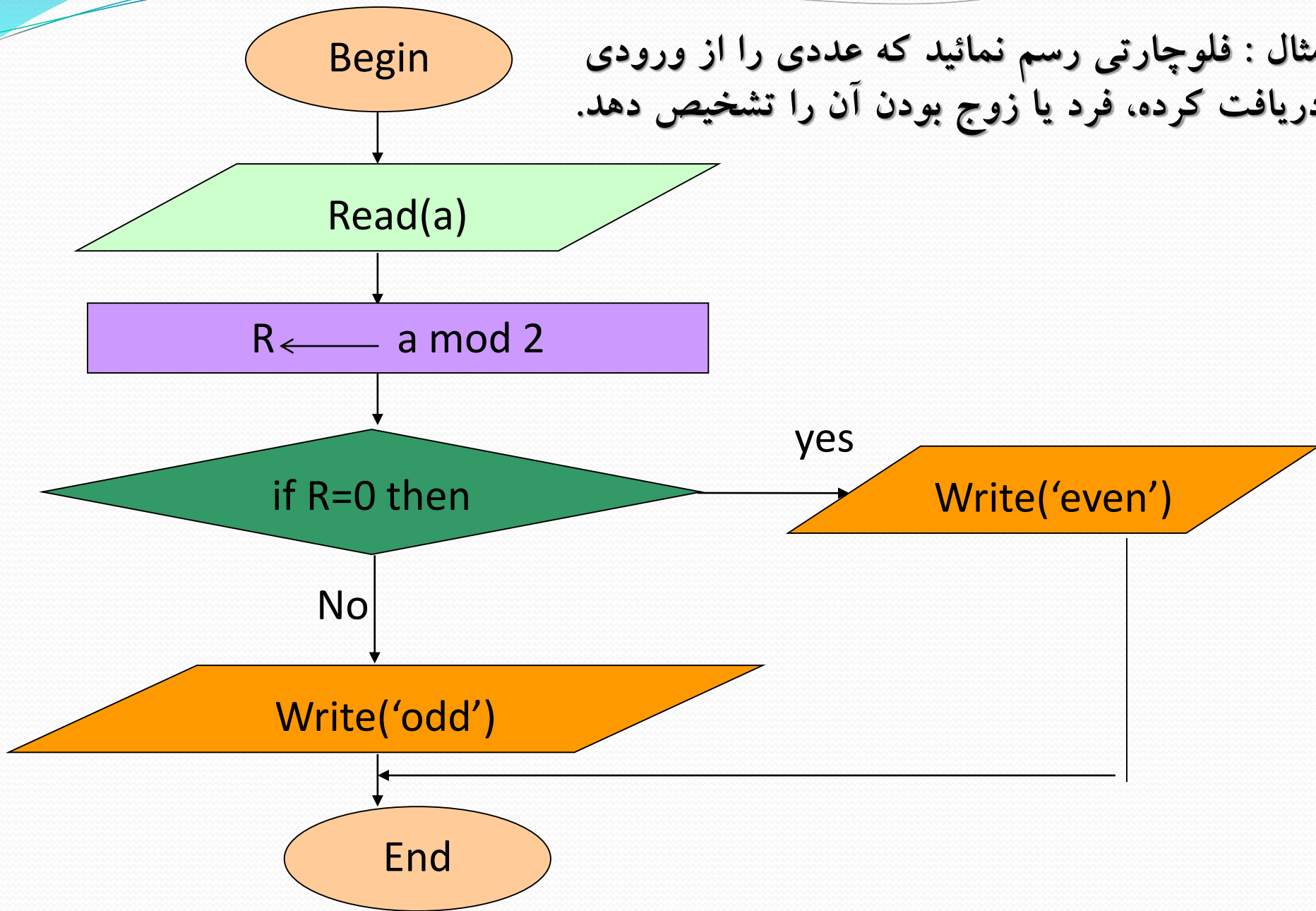
در حل بسیاری از مسائل یا تقریباً تمام مسائل نیاز به استفاده از شروط جزء، نیازهای اساسی محسوب می‌شود. همانطور که ما خودمان در زندگی روزمره با این شرط‌ها سرکار داریم. بطور مثال اگر هوا ابری باشد ممکن است چنین سخن بگوییم:

اگر هوا بارانی باشد سپس چتری برمی‌دارم. در غیر اینصورت چتر برنمی‌دارم.

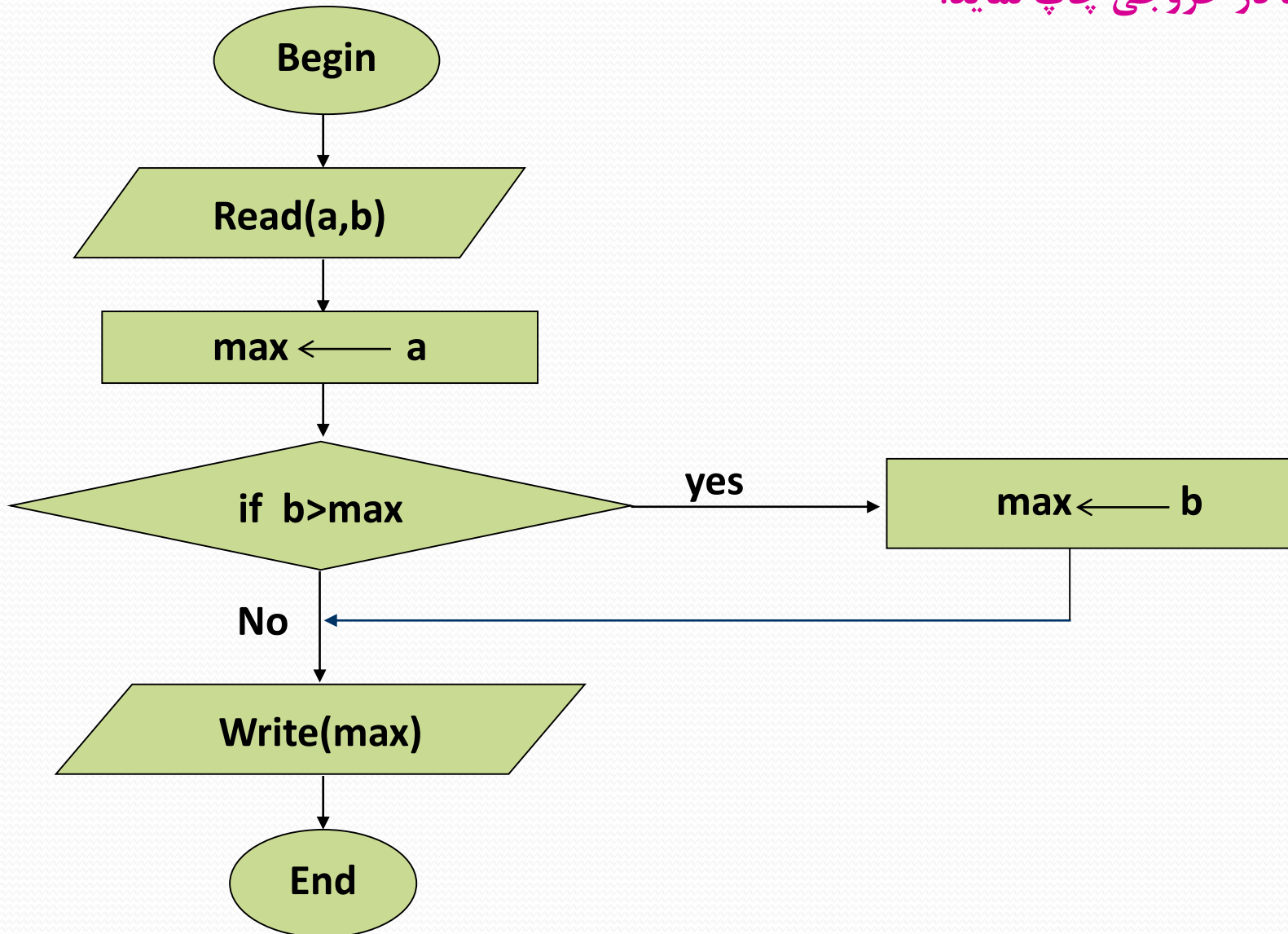
در حالت کلی شرط را بصورت زیر نمایش می دهند:



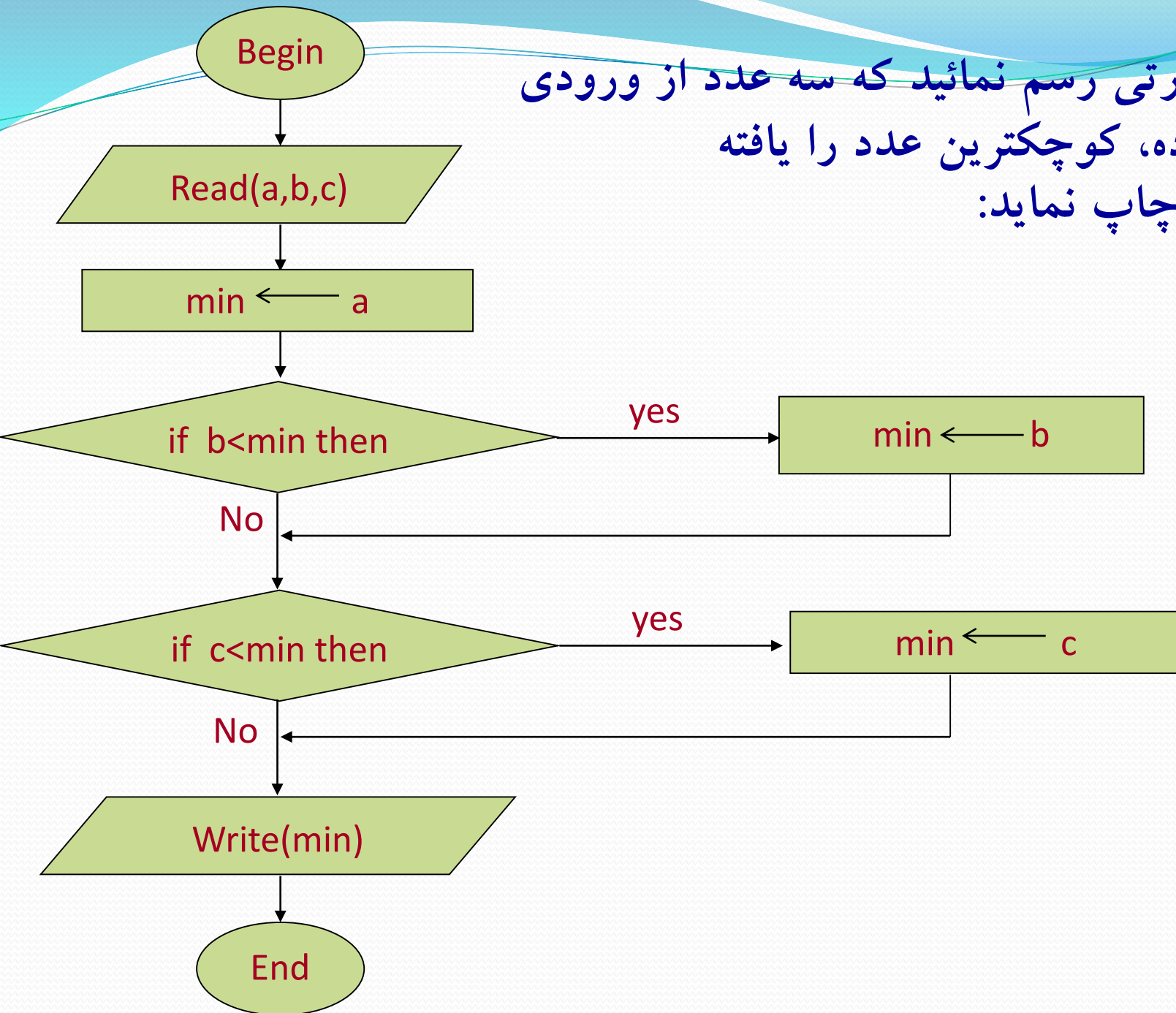
مثال : فلوچارتی رسم نمائید که عددی را از ورودی دریافت کرده، فرد یا زوج بودن آن را تشخیص دهد.



مثال : فلوچارتی رسم کنید که دو عدد از ورودی دریافت کرده بزرگترین عدد را پیدا کرده در خروجی چاپ نماید.



مثال : فلوچارتی رسم نمائید که سه عدد از ورودی دریافت کرده، کوچکترین عدد را یافته در خروجی چاپ نماید:



نمونه اجرای فلوچارت بالا بصورت زیر می باشد:

	a	b	c	Min	خروجی
1	12	11	17		11
2				12	
3				11	
4				11	
5				11	

تمرین سری دوم

- ۱- فلوچارتی رسم کنید که عددی را از ورودی دریافت کرده، قدر مطلق عدد را در خروجی چاپ کند.
- ۲- فلوچارتی رسم نمائید که عددی از ورودی دریافت کرده مثبت، منفی یا صفر بودن عدد را تشخیص داده، در خروجی با پیغام مناسب چاپ کند.
- ۳- فلوچارتی رسم نمائید که عددی را از ورودی دریافت کرده، بخشپذیری آن بر ۳ و ۵ را بررسی نماید.
- ۴- فلوچارتی رسم نمائید که ضرایب یک معادله درجه دوم را از ورودی دریافت کرده، ریشه‌های آن را محاسبه در خروجی چاپ کند.

❖ حلقه‌ها

در حل بسیاری از مسائل با عملیاتی روبرو می‌شویم ، که نیاز به تکرار دارند و عمل تکرار آنها به تعداد مشخصی انجام می‌گیرد. فرض کنید،

بخواهیم میانگین ۱۰۰ عدد را محاسبه کنیم، در اینصورت منطقی بنظر نمی‌رسد که ۱۰۰ متغیر مختلف را از ورودی دریافت کنیم سپس آنها را جمع کنیم.

انواع حلقه ها

حلقه های با تکرار مشخص

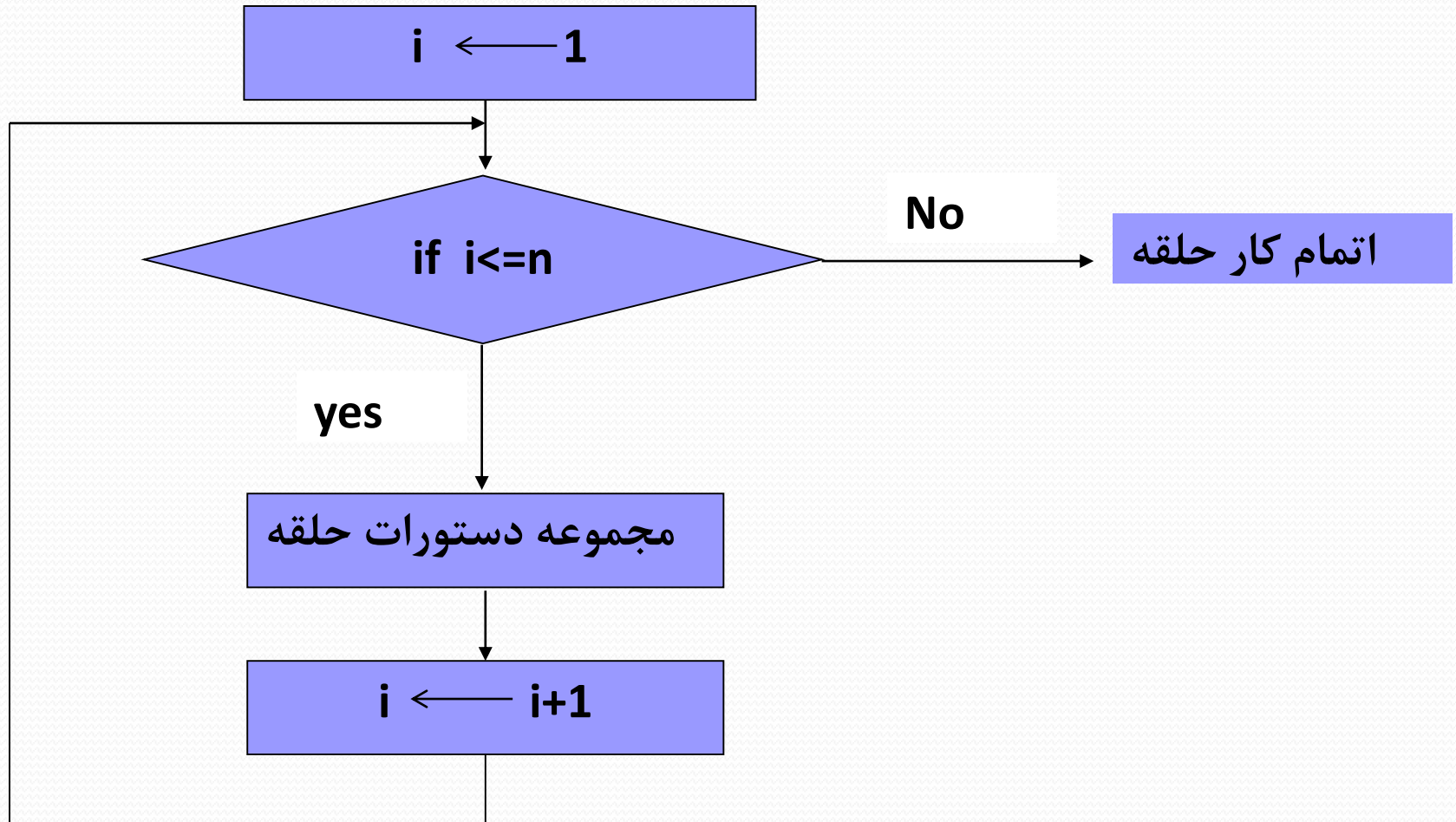
حلقه های با تکرار نامشخص

حلقه های با تکرار مشخص

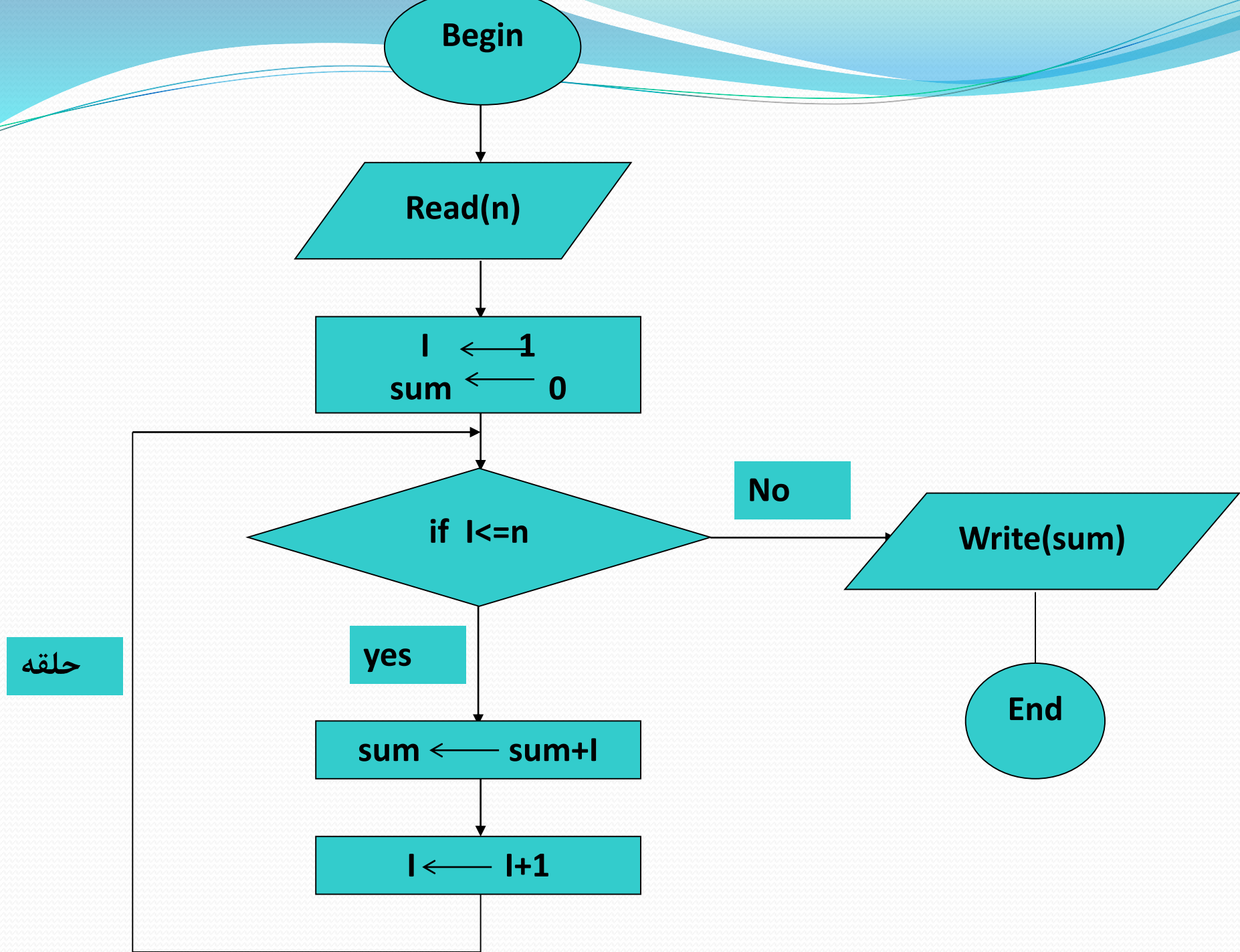
در این نوع حلقه ها تعداد تکرار مشخص می باشد این حلقه از اجزاء زیر تشکیل می شود:

- ۱- اندیس حلقه
- ۲- مقدار اولیه برای اندیس حلقه
- ۳- مقدار افزاینده برای اندیس حلقه (معمولا یک واحد در هر مرحله)
- ۴- مقدار نهایی (تعداد تکرار حلقه)
- ۵- شرطی برای کنترل تعداد تکرار حلقه

این حلقه‌ها را غالباً با فلوچارت بصورت زیر نمایش می‌دهند:



مثال : فلوچارتی رسم نمائید که عدد n را از ورودی دریافت کرده،
مجموع اعداد از یک تا n را محاسبه کند.



نمونه اجرای فلوچارت بالا بصورت زیر است:

	N	I	sum	خروجی
1	2			3
2		1	0	
3			1	
4		2		
5			3	
6		3		
7				

مثال : فلوچارتی رسم کنید که n عدد از ورودی دریافت کرده،
بزرگترین مقدار از بین n عدد را پیدا کرده در خروجی چاپ نماید.

Begin

Read(n)
Read(a)

$i \leftarrow 2$
 $\text{max} \leftarrow a$

if $i \leq n$ then

No

write(max)

End

yes

Read(a)

if $a > \text{max}$

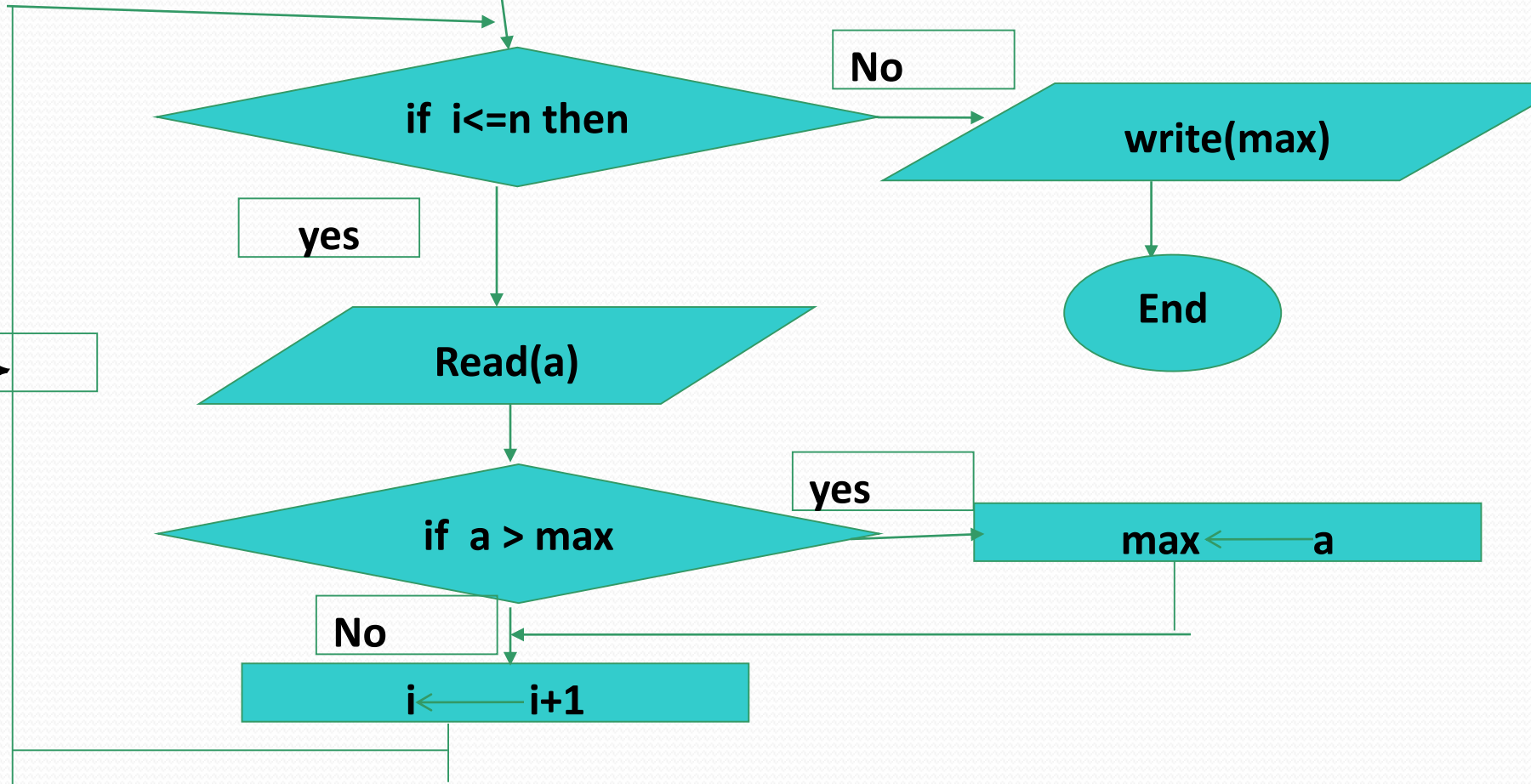
yes

$\text{max} \leftarrow a$

No

$i \leftarrow i + 1$

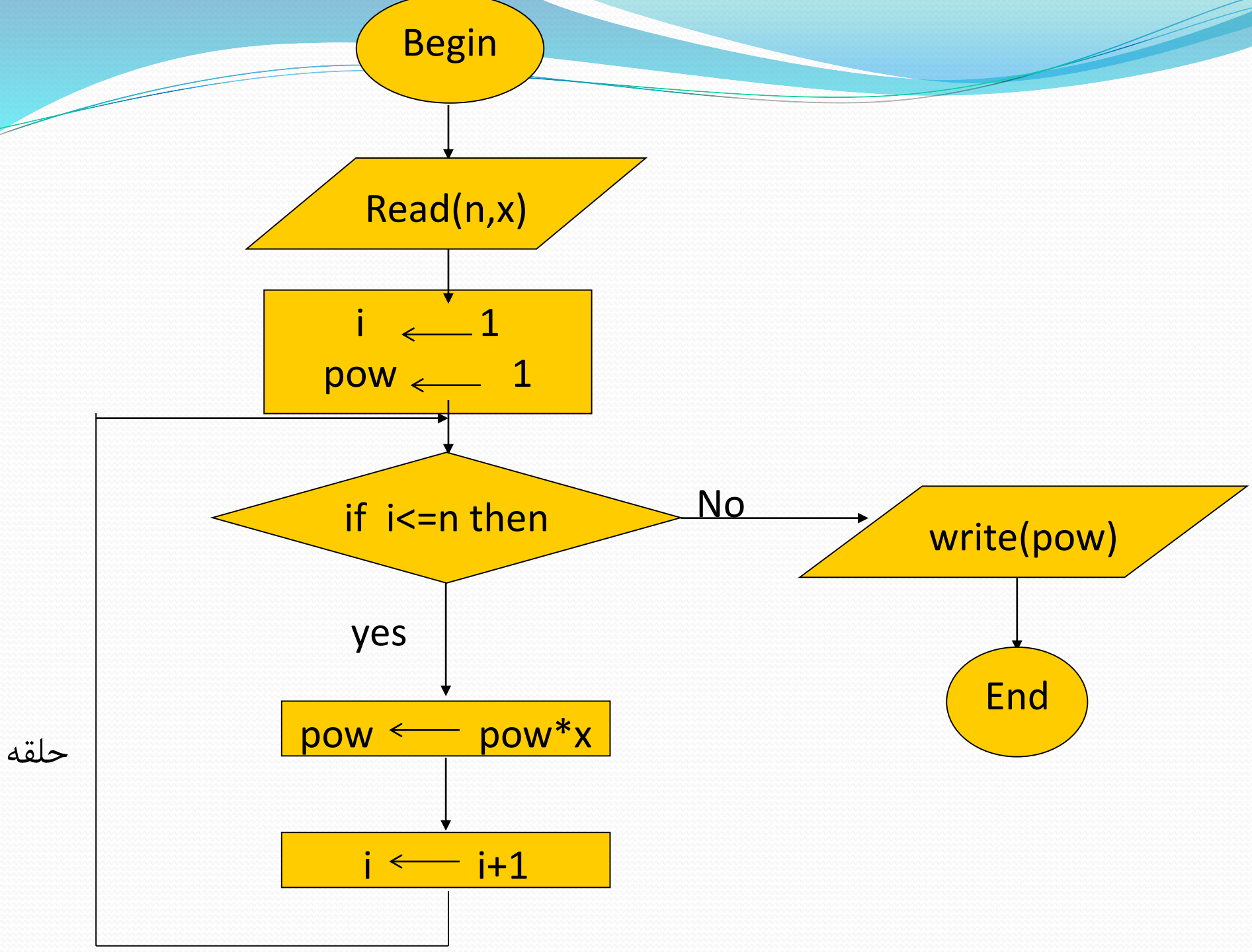
حلقه



مثال : فلوچارتی رسم نمائید که x , n ، دو عدد صحیح مثبت را از ورودی دریافت کرده سپس x به توان n را محاسبه کند.

x^n

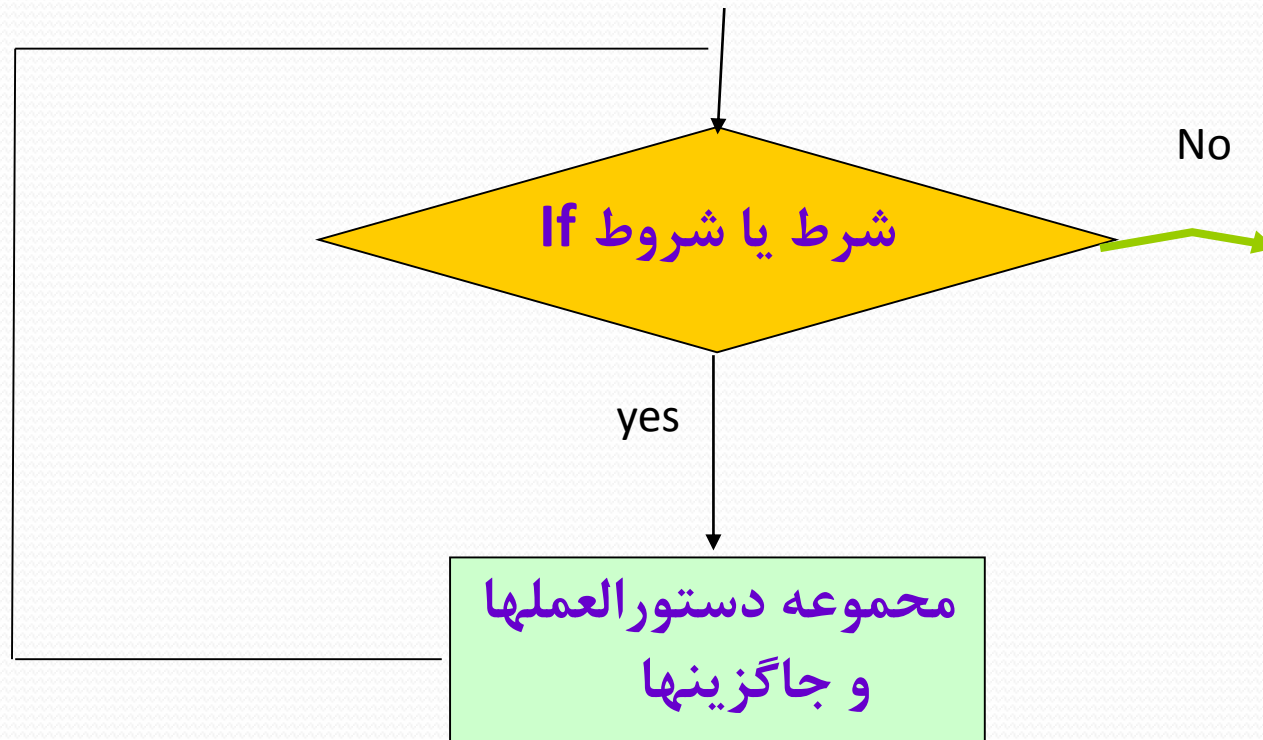
اندیس حلقه
مقدار نهایی



- حلقه‌هایی که تعداد تکرار آنها مشخص نیست

در این حلقه‌ها با توجه به ورودی، تعداد تکرار مشخص می‌شود و دقیقاً نمی‌توان تعداد تکرار حلقه را بدون ورودی معین کرد. این حلقه‌ها فقط شامل شرطی هستند که تا زمانی که برقرار باشد حلقه اجرا می‌شود.

در حالت کلی این نوع حلقه‌ها بصورت زیر نمایش داده می‌شوند:



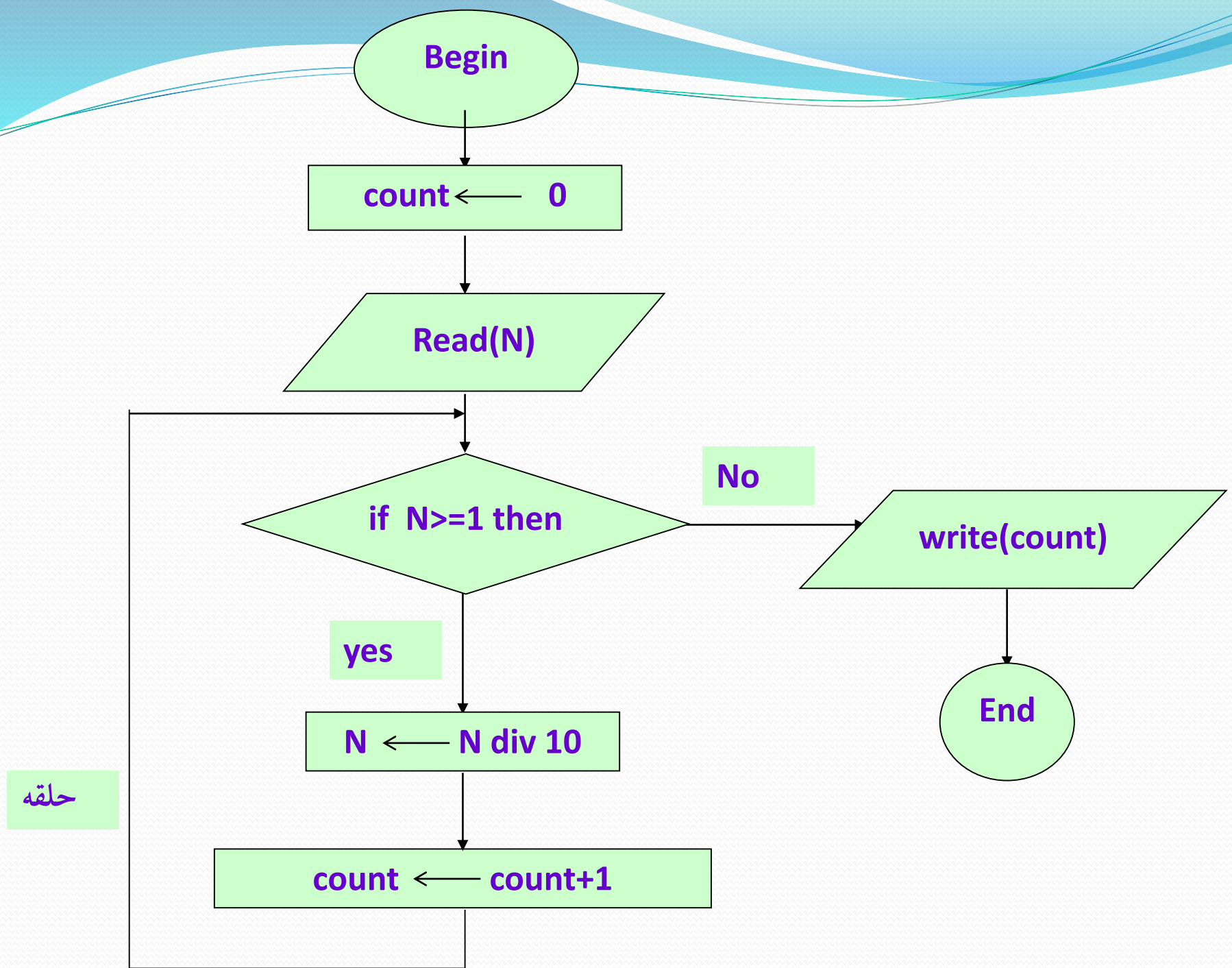
مثال: فلوچارتی رسم کنید که عددی را از ورودی دریافت کرده سپس تعداد ارقام آن را شمرده در خروجی چاپ نماید.

N

عدد خوانده شده

count

تعداد ارقام

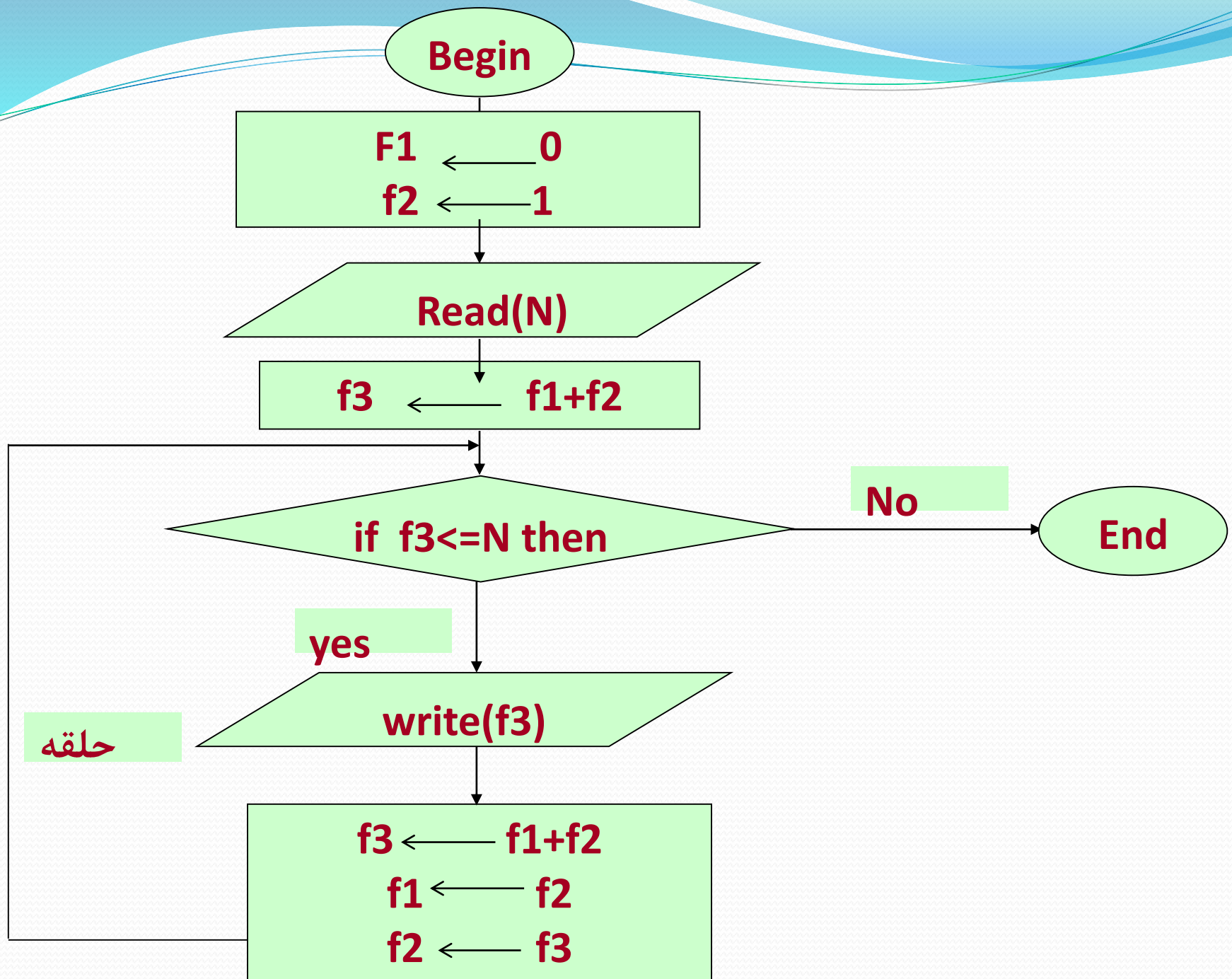


مثال : فلوچارتی رسم نمائید که عددی از ورودی دریافت کرده، سری فیبوناتچی قبل از آنرا تولید نماید.

در حالت کلی جملات سری بصورت:

$$f_k = f_{k-1} + f_{k-2}$$

N	عدد خوانده شده
f_1	جمله اول سری
f_2	جمله دوم سری
f_3	جمله سوم سری



تمرین سری سوم

۱- فلوچارتی رسم نمائید که عددی از ورودی دریافت کرده، کامل بودن آنرا بررسی نماید. (عدد کامل، عددی است که مجموع مقسوم‌علیه‌های آن با خودش برابر باشد.)

۲- فلوچارتی رسم کنید که N را از ورودی دریافت کرده، N جمله سری فیبوناچی را تولید نماید.

۳- فلوچارتی رسم نمائید که دو عدد M , N را از ورودی خوانده، بزرگترین مقسوم‌علیه مشترک دو عدد را محاسبه و چاپ کند.

❖ حلقه‌های تودرتو

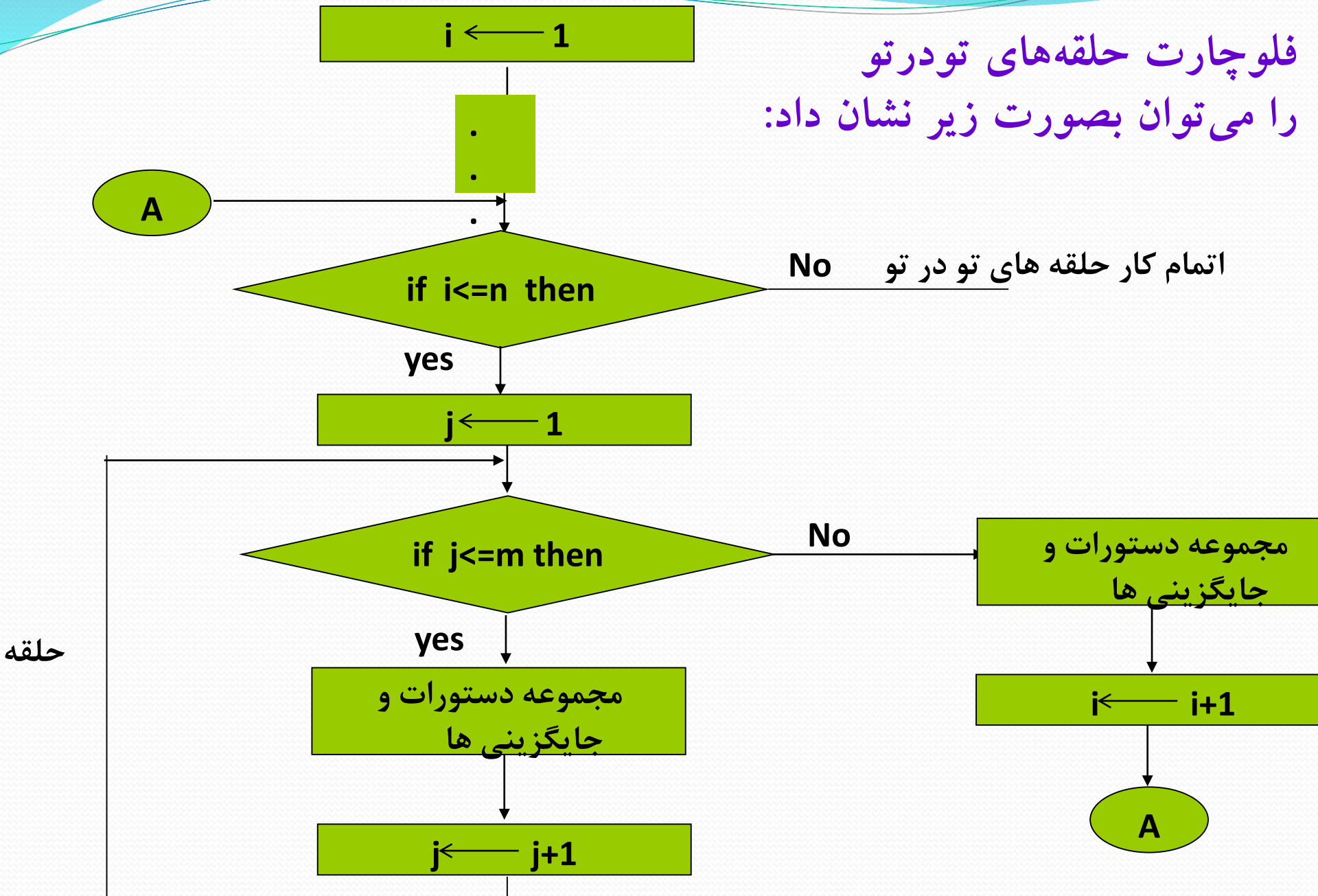
الگوریتم‌هایی که تا حال بکار بردیم، فقط شامل یک حلقه بودند. در صورتی که در بسیاری از مسائل ممکن است نیاز به استفاده از چند حلقه در داخل هم باشیم. در این نوع حلقه‌ها باید دقت بیشتری به خرج دهیم، تا مشکلی پیش نیاید. اگر از حلقه‌های نوع اول بصورت تودرتو استفاده کنیم در اینصورت برای هر حلقه شرط نهایی و اندیس اولیه جداگانه باید تعریف کنیم.

در حلقه‌های تو در تو به ازای یکبار تکرار حلقه اولیه، حلقه داخلی به اندازه مقدار نهایی خود تکرار می‌شود. در کل اگر حلقه اولیه n بار تکرار شود و

حلقه داخلی m بار، در این صورت کل حلقه : $n \times m$

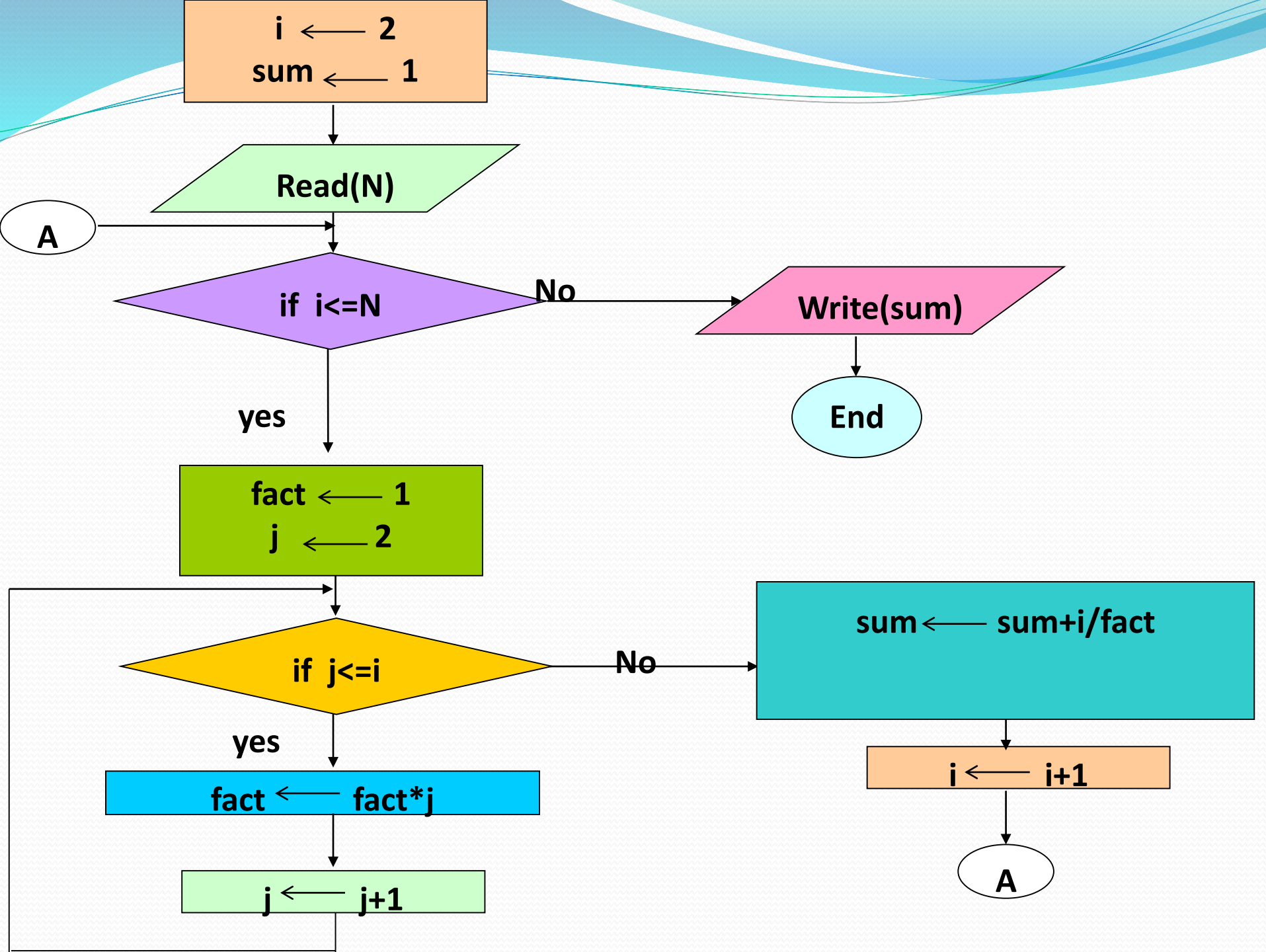
بار تکرار خواهد شد.

فلوچارت حلقه‌های تو در تو
را می‌توان بصورت زیر نشان داد:

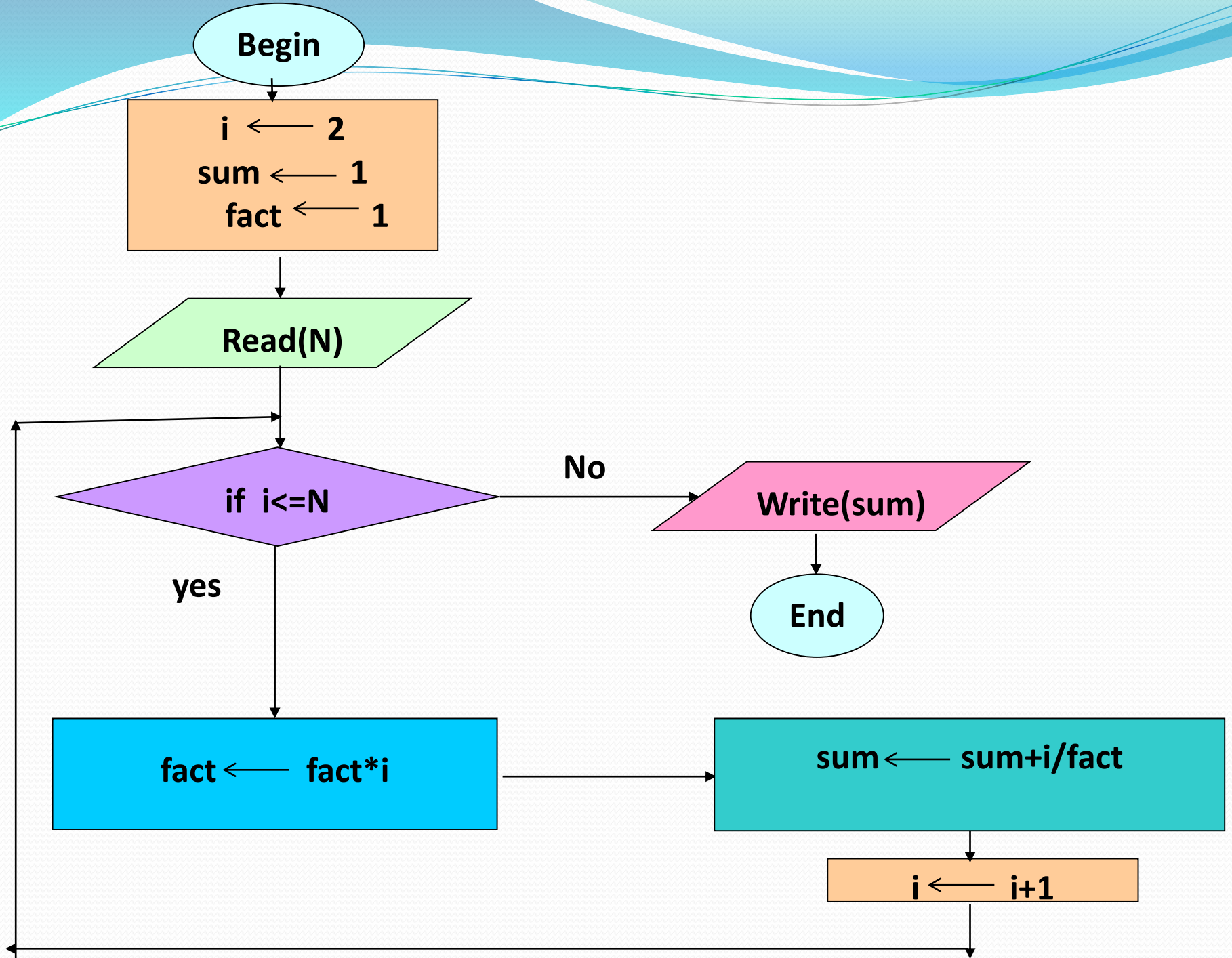


مثال : فلوچارتی رسم نمائید که **N** را از ورودی دریافت کرده، مجموع سری زیر را محاسبه نماید:

$$S = 1 + \frac{2}{2!} + \frac{3}{3!} + \dots + \frac{N}{N!}$$



[illegible]



❖ تمرینات آخر فصل

۱- فلوچارتی رسم نمائید که N عدد از ورودی دریافت کرده تعداد اعداد اول و کامل را شمرده در خروجی چاپ نماید.

۲- فلوچارتی رسم نمائید که N, X را از ورودی خوانده مقدار سری زیر را محاسبه کند:

$$S = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^N}{N!}$$

۳- فلوچارتی رسم نمائید که عددی را از ورودی دریافت کرده مقلوب عدد را محاسبه و در خروجی چاپ کند.

۴- فلوچارتی رسم کنید که تاریخ تولد شخصی را از ورودی خوانده، سن شخص را با تاریخ روز، محاسبه نموده در خروجی چاپ کند.

۵- فلوچارتی رسم نمائید که N, M (که $m > n$) را از ورودی دریافت کرده سری فیبوناچی بین N, M را تولید کرده، در خروجی چاپ کند.

کاربرد آرایه ها در الگوریتم ها

هدفهای کلی

- شناخت آرایه ها و مفهوم آن
- شناخت الگوریتم های لازم برای جستجو و مرتب سازی
- مقایسه انواع روش های جستجو با هم

مقدمه

فرض کنید بخواهیم اطلاعات ۱۰۰ کارمند را از ورودی بخوانیم و سپس آنها را مرتب کنیم، در اینصورت باید ورودیها را در جایی از حافظه ذخیره کنیم. در زبانهای برنامه‌نویسی معمولاً از آرایه برای ذخیره اطلاعات در حافظه استفاده می‌کنند. در آرایه‌ها ما با توجه به تعداد ورودیها، طول آن را مشخص می‌کنیم. سپس داده‌ها را خوانده در آن قرار می‌دهیم.

❖ تعریف آرایه

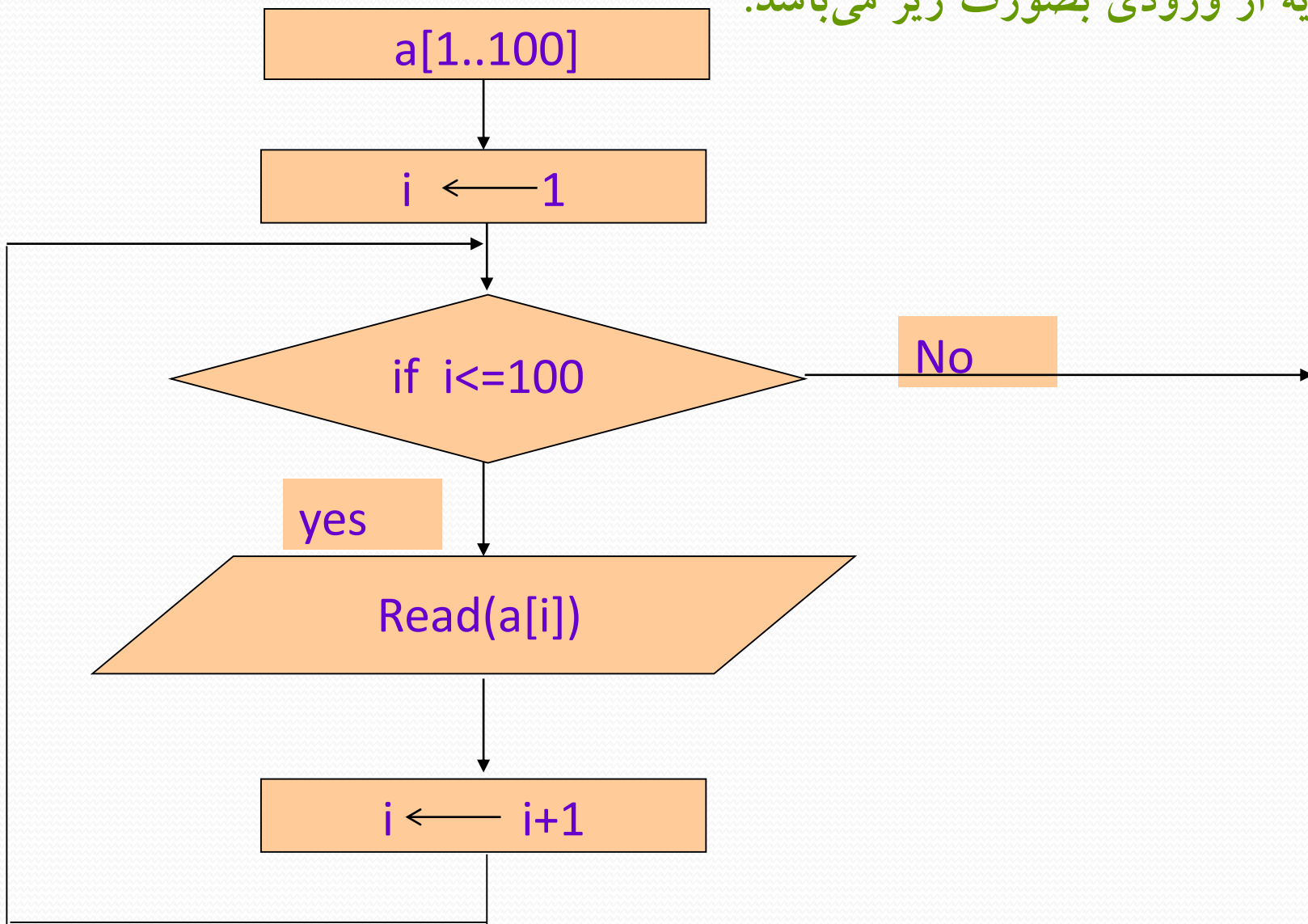
خانه‌های پشت سر هم از حافظه، که هم‌نوع بوده و توسط یک اسم معرفی می‌شوند، آرایه نام دارد. نحوه دسترسی به هر یک از اعضاء آرایه، از طریق اندیس آرایه امکانپذیر است. برای تعریف آرایه ابتدا طول آرایه که در حقیقت تعداد خانه‌های آن را مشخص می‌کند، معین می‌کنیم. سپس نوع خانه‌ها باید معین شوند.

در فلوچارت‌ها آرایه‌ها را بصورت زیر نمایش می‌دهیم:

Name[1 .. Length]

طول آرایه اسم آرایه

برای خواندن یک آرایه از ورودی از حلقه‌ها استفاده می‌کنیم. فلوچارت
خواندن آرایه از ورودی بصورت زیر می‌باشد:

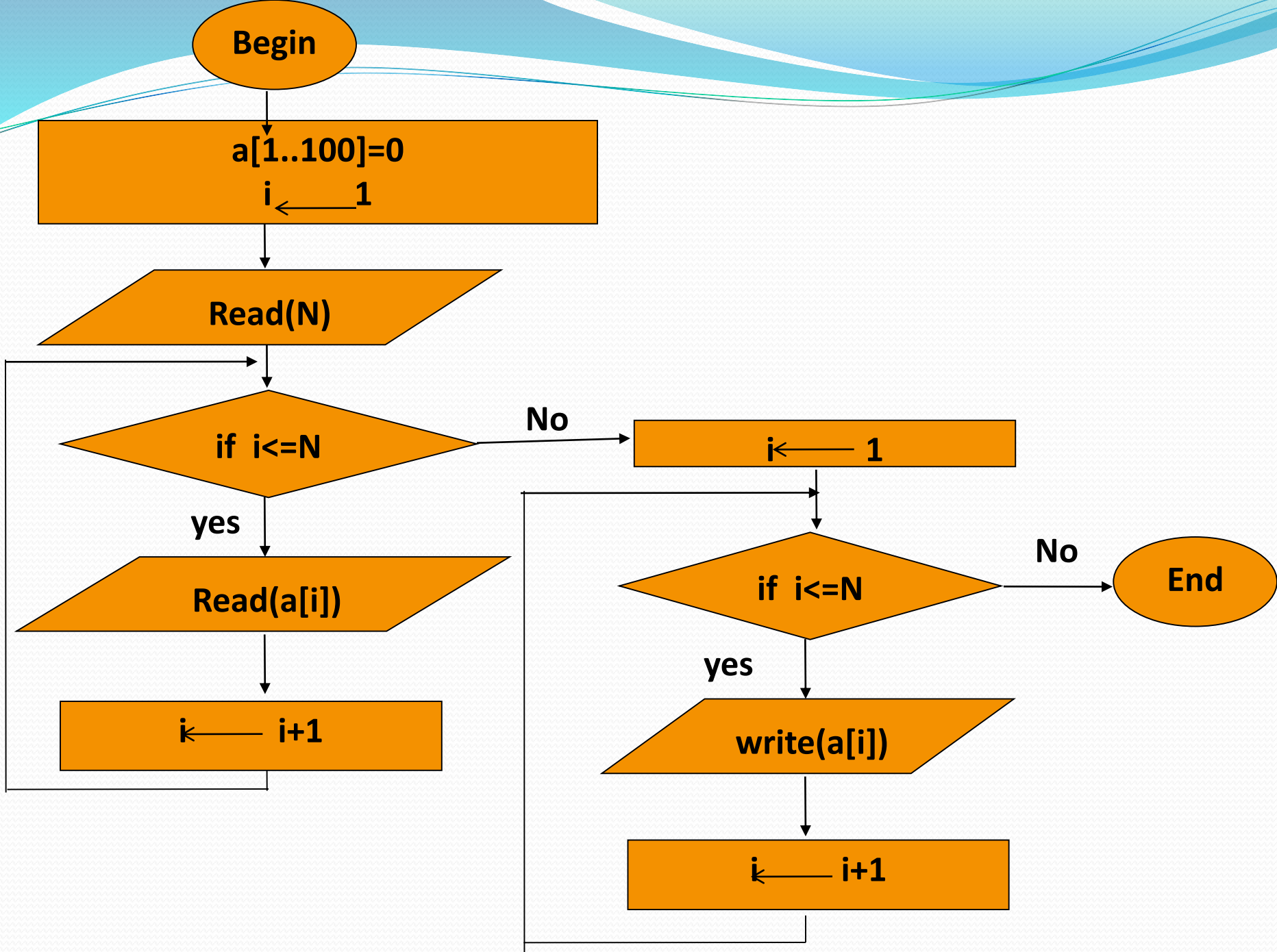


با توجه به فلوچارت بالا برای دسترسی به عنصر نام آرایه در حالت کلی بصورت عمل می کنند :

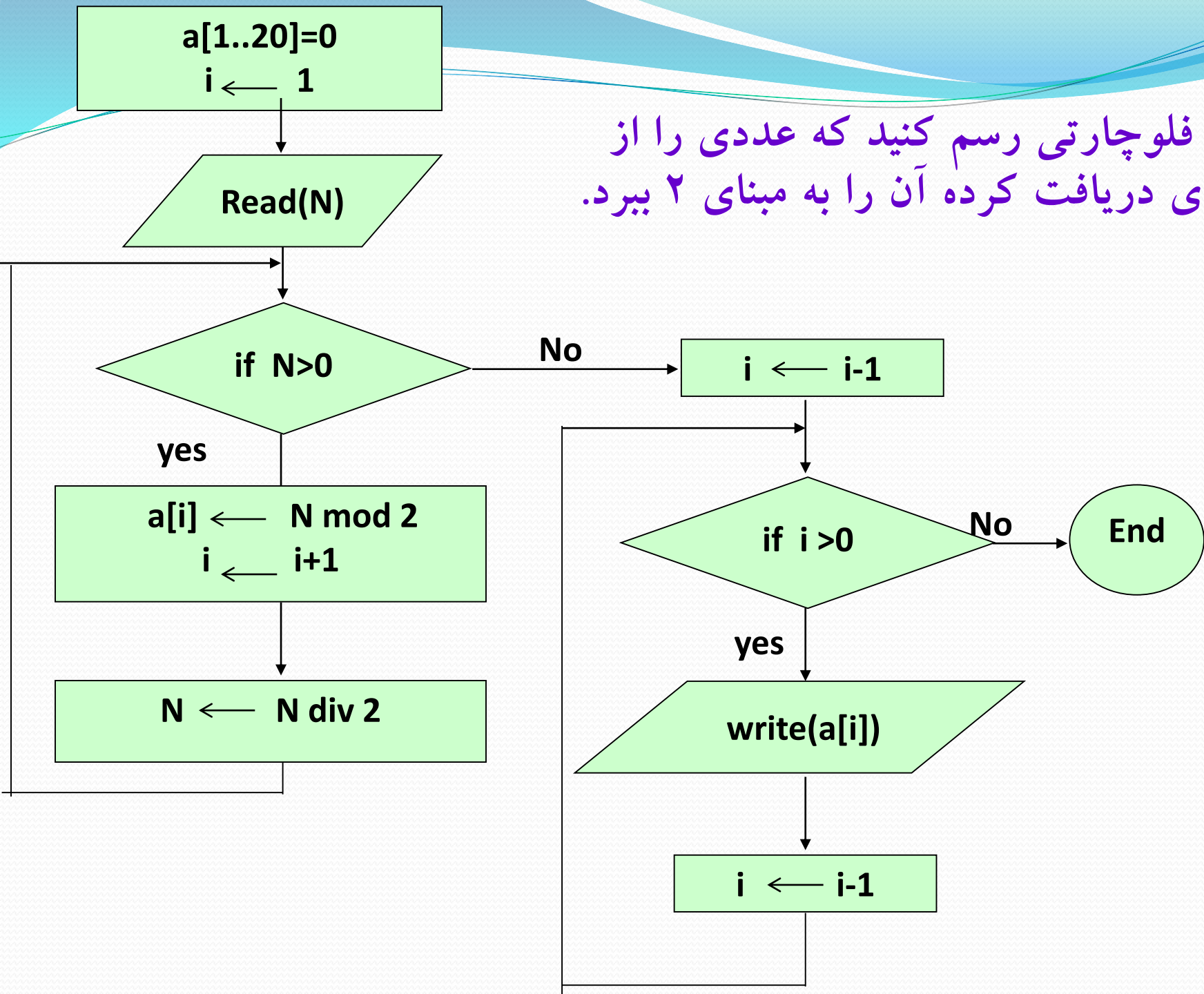
Name [index]

اندیس آرایه اسم آرایه

مثال: فلوچارتی رسم کنید که یک آرایه حداکثر ۱۰۰ عنصری را از ورودی دریافت کرده، سپس آن را خروجی نمایش دهد.



مثال : فلوجارتي رسم كنيد كه عددي را از
ورودي دريافت كرده آن را به مبناي ۲ ببرد.



Parameter \ No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
N			3				1				0									
a[1..20]	0				a[1]=1				a[2]=1											
i		1				2				3			2			1			0	
IF				Y				Y				N		Y			Y			N
Output															1			1		

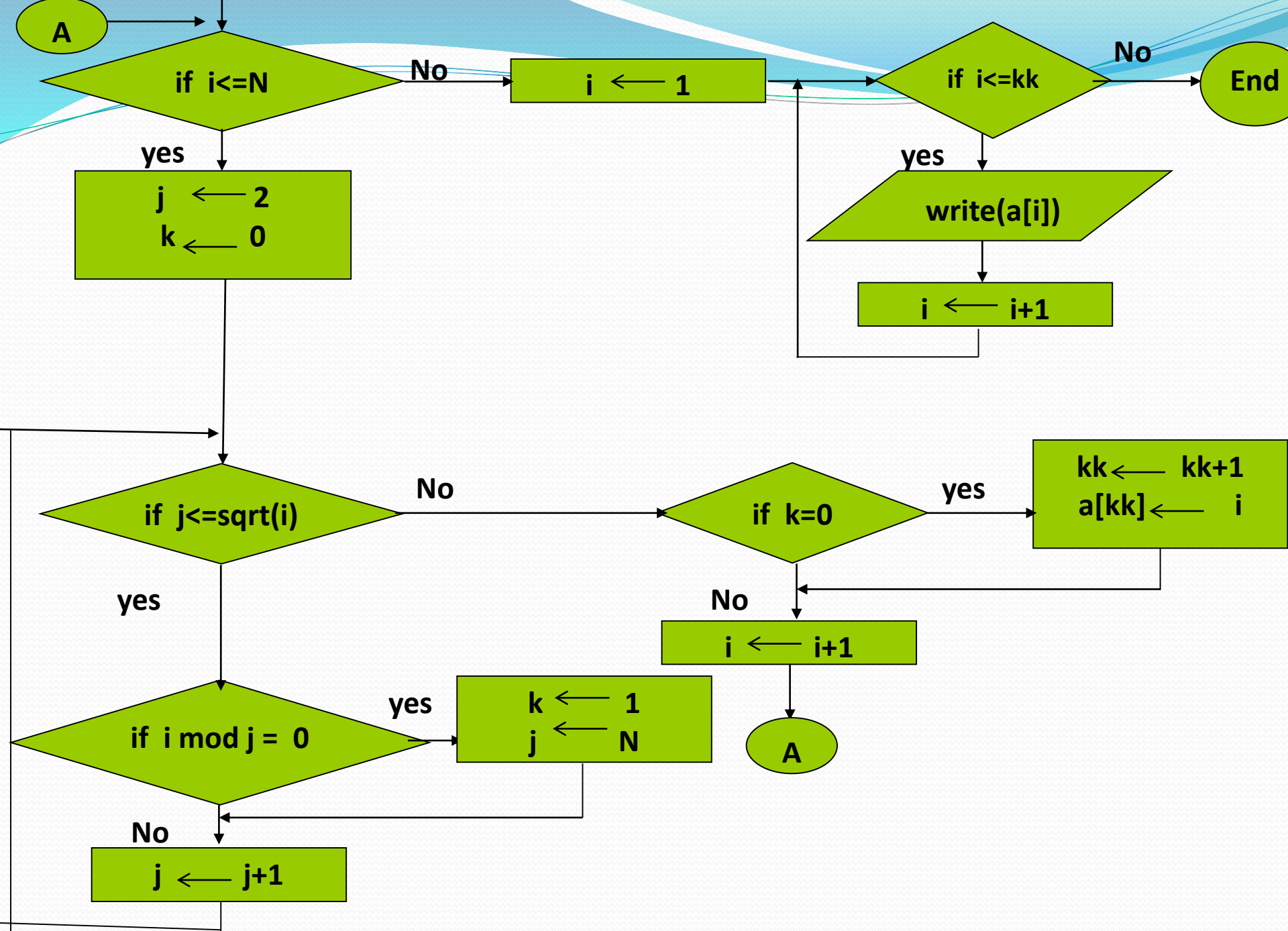
تمرین : فلوچارتی رسم نمائید که عددی از ورودی دریافت کرده سپس اعداد اول قبل از آن را تولید نموده ، در یک آرایه قرار دهد.

$a[1..100]=0$

$i \leftarrow 2$

Read(N)

$kk \leftarrow 0$



❖ جستجو و مرتب سازی (search and sort)

یکی از مسائلی که در بحث طراحی الگوریتم بسیار مهم است، بحث مرتب سازی و جستجو می باشد. منظور از جستجو اینست که یک مقداری را از یک لیست جستجو کنیم و منظور از مرتب سازی اینست که یک لیست مرتب از داده ها را تولید کنیم.

برای جستجو و مرتب سازی الگوریتم های مختلفی وجود دارد در زیر الگوریتم های اولیه، برای جستجو و مرتب سازی را بررسی می کنیم.

❖ الگوریتم های جستجو (search)

دو الگوریتم زیر غالباً برای جستجو بکار می روند:

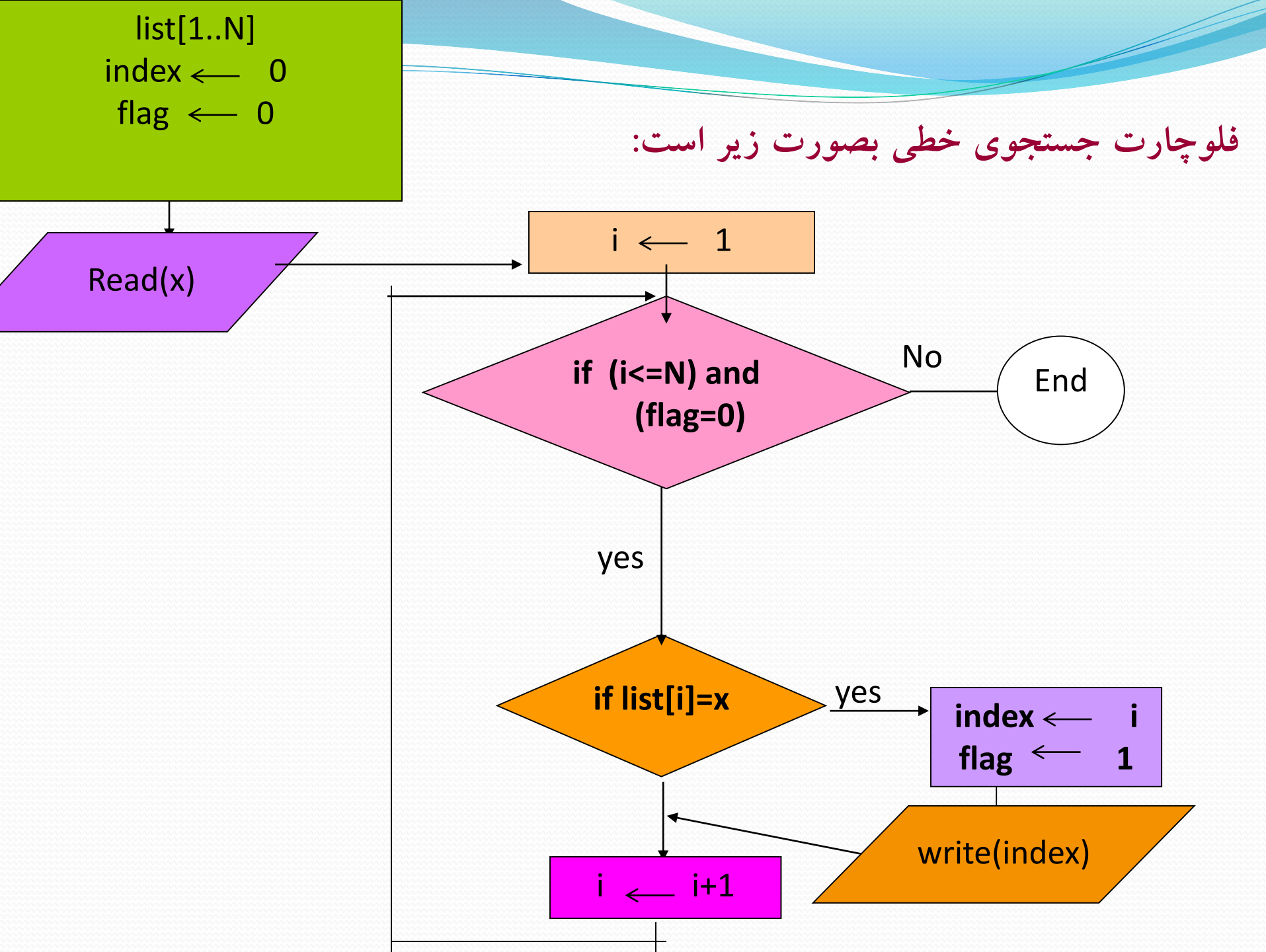
linear search جستجوی خطی

binary search جستجوی دوتائی

جستجوی خطی *linear search*

در جستجوی خطی عبارت مورد جستجو را به ترتیب با اولین، دومین و ... عنصر آرایه مقایسه می کنیم اگر عنصر مورد جستجو پیدا شد اندیس آن را نمایش می دهیم.

فلوچارت جستجوی خطی بصورت زیر است:



جستجوی دوتایی *binary search*

در جستجوی دوتایی ، لیست مورد جستجو، مرتب می باشد. لذا برای جستجو اعمال زیر انجام می شود:

۱- عنصر X با عنصر وسط آرایه که اندیس آن برابر

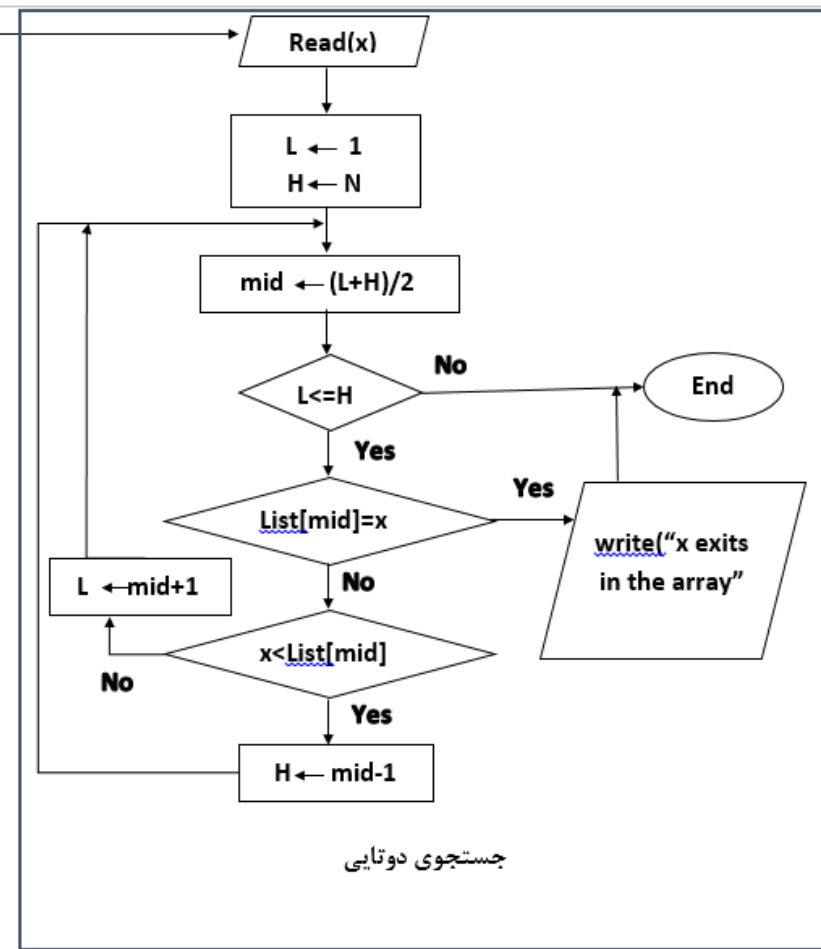
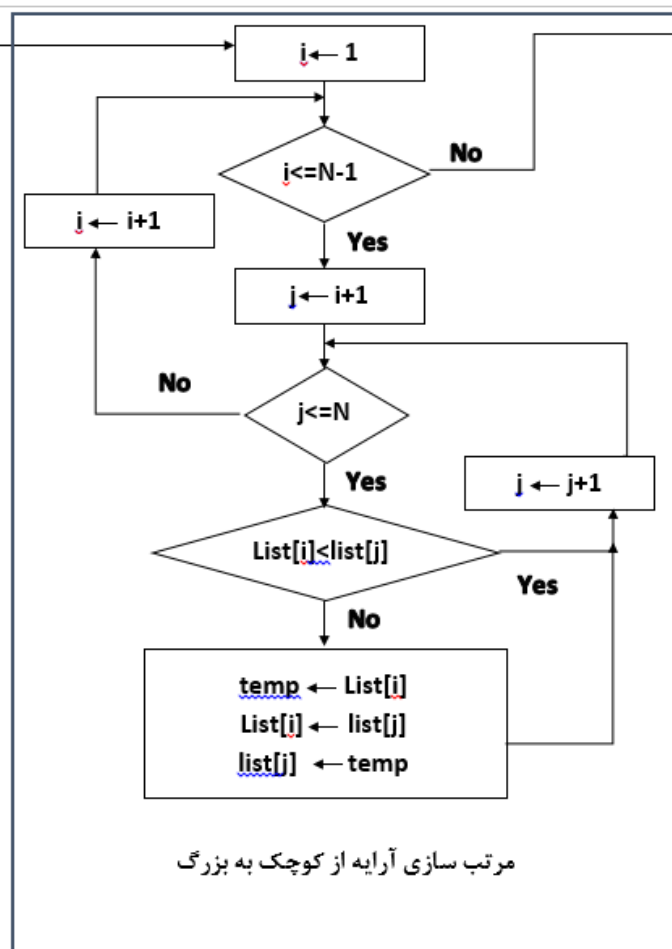
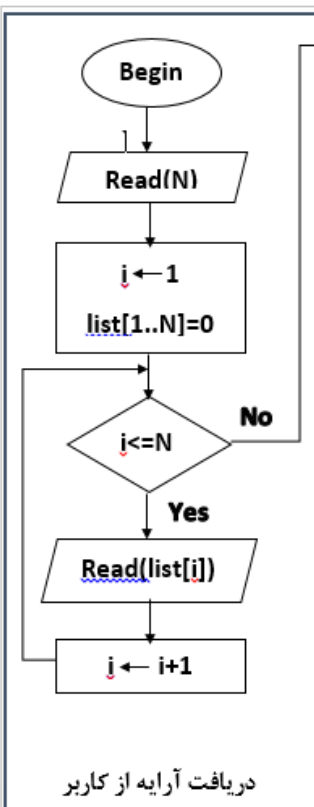
$\text{middle} \leftarrow (\text{low} + \text{high}) / 2$

مقایسه می شود.

۲- اگر x از عنصر وسط کوچکتر باشد، عنصر مورد نظر احتمالاً در قسمت بالای لیست قرار دارد. لذا آرایه با اندیس، جدید در نظر گرفته می شود و قسمت پایین لیست از فضای جستجو حذف می شود.

۳- اگر x از عنصر وسط بزرگتر باشد قسمت بالای لیست حذف می شود و فضای جستجو، قسمت پایین آرایه خواهد بود.

۴- اگر x برابر عنصر وسط باشد عمل جستجو خاتمه می پذیرد.



تمرین

۱- فلوجارتی رسم نمائید که عددی از ورودی دریافت کرده، اعداد کامل قبل از خود را تولید و در یک آرایه قرار دهد.

۲- فلوجارتی رسم نمائید که یک آرایه حداکثر ۱۰۰ عنصری از ورودی دریافت کرده، عناصری از آن که اول هستند را با صفر کردن حذف نماید.

۳- فلوجارتی رسم نمائید که یک عدد حداکثر ۲۰ رقمی را توسط آرایه‌ای از ورودی دریافت نماید. سپس یک عدد تک رقمی را از ورودی خوانده در عدد ۲۰ رقمی ضرب نموده، حاصل را در خروجی چاپ نماید.