به نام خدا

دانشکده فنی امام‌علی(ع) کردکوی

مبانی کامپیوتر و برنامه نویسی

مدرس : محمد جلالی

تذکر: دانشجویان توجه داشته باشند کلیه برنامه ها و نکاتی که در کلاس مطرح می شود جزء امتحان پایان ترم محسوب خواهد شد.

Computer

محسابه گر

تعريف كامپيوتر:

يک دستگاه الکتريکی و الکترومکانيکی که داراي حافظه و قابل برنامه ريزي می باشد و قادراست عمليات رياضي و منطقي را با سرعت بالا انجام دهد.

سيستم:

مجموعه عناصر و اجزاء مرتبط به هم، که در مجموع هدف خاصي را دنبال مي کنند.

Computer محسابه گر

کامپیوتر:

توانایی انجام کارهای زیر را دارد:

۱. دریافت ورودی ۲. پردازش اطلاعات ۳. بازیابی اطلاعات

مزاياي كامپيوتر نسبت به انسان:

۱- سرعت بالا

۲- دقت بالا

۳- حافظه بالا

مزاياي انسان به كامپيوتر:

۱- خلاق و قادر به ابداع و اختراع

۲- در مواجه با مسائل جديد انعطاف پذير و تصميم گيرهستند

۳- دارای عواطف و درک متقابل

به طور کلی علوم کامپیوتر را به دو گروه اصلی تقسیم میکنند

۱- **hardware سخت افزار:** به کلیه قطعات فیزیکی و قابل لمس کامپیوتر سخت افزار گفته میشود.

۲- **software نرم افزار:** به کلیه برنامه ها و دستورالعمل هایی که جهت ارتباط با کامپیوتر و استفاده از آن به کار می رود.

firm ware میان افزار: تجهیزاتی که از ترکیب سخت افزار ونرم افزار به وجود می آید. مانند: Rom

انواع کامپیوترها

الف – ریزکامپیوترها (micro computer) : کوچکترین وارزانترین نوع کامپیوترها که بصورت خانگی وشخصی استفاده می شوند. به همین خاطر به آنها کامپیوترهای شخصی PC (Personal Computer) می گوییم. انواع آن:

۱- رومیزی (desktop)

۲- کیفی (Laptop یا Notebook)

۳- کامپیوترهای دستی palmtop

۴- مشاور دیجیتال شخصی (PDA: Personal Digital Assistants)

مقایسه انواع کامپیوترها

نوع کامپیوتر	قابلیت حمل	قابلیت شارژ باطری	نوع مونتاژ	کارایی	قیمت
رومیزی	✗	✗	مونتاژ توسط فرد	متوسط	متوسط
کیفی	✓	✓	مونتاژ توسط شرکت Q.C زیر نظر واحد کنترل کیفیت	خوب	بالاتر

انواع کامپیوترها از لحاظ وسعت تجهیزات و قدرت پردازش

ب- کامپیوترهای کوچک (mini computer):
درمراکز اداری وتجاری و دانشگاهی

ج- کامپیوترهای بزرگ یا متوسط (mainframe computer):
مراکز بزرگ اداری و دولتی و وزارتخانه ها

د- ابرکامپیوترها (supercomputer):
استفاده در مراکز استراتژیک. فقط چند تا در جهان وجود دارد. (نظامی، فضایی و اتمی)

Data

داده ها: هرگونه اطلاعاتي كه مي تواند در كامپيوتر مورد استفاده قرار گيرد.

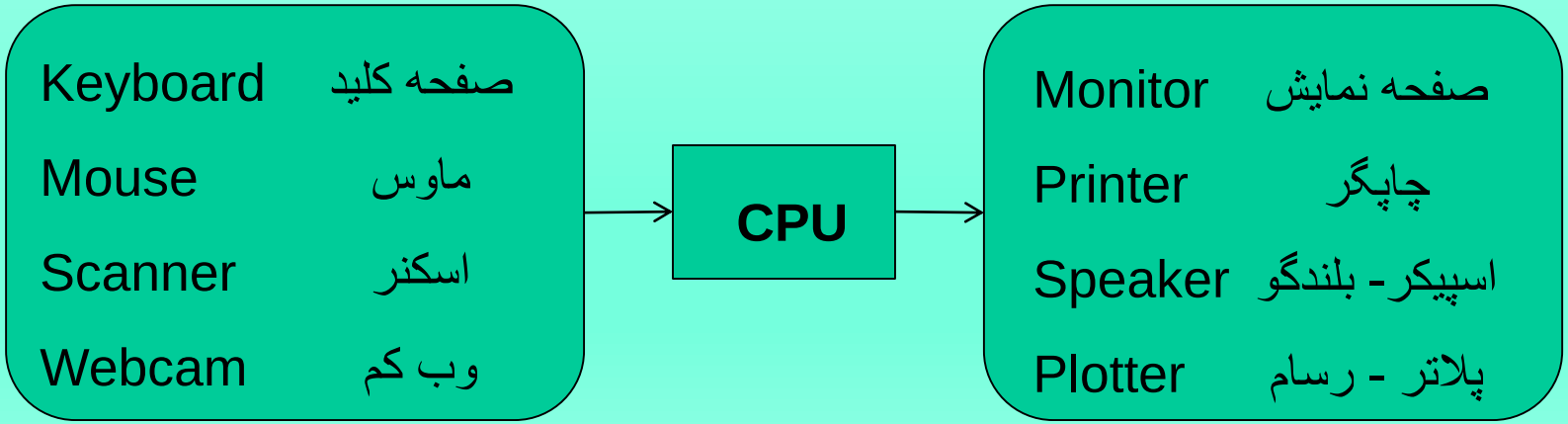
انواع داده ها:

- متني
- تصويري
- صوتي
- و...

قطعات کامپیوتر

قطعاتي كه داده ها را از واحد پردازش دريافت كرده و به فرم نمايشي، چاپي و صوتي تبديل مي كنند.

قطعاتي كه داده ها را دريافت كرده و به بخش پردازش ارسال مي كنند.



قطعات کامپیوتر

قطعات مقابل ورودي اند يا خروجي؟

CD Writer

DVD Writer

F.D.D

Modem

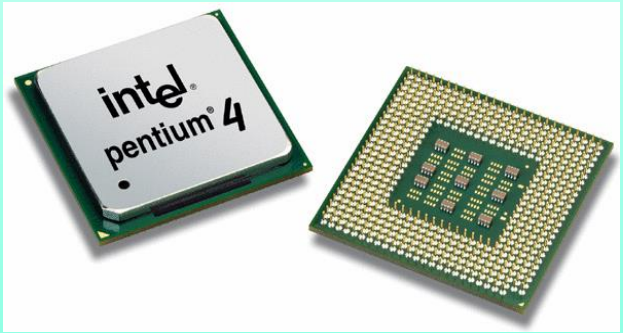
Network Card

اين قطعات هم به عنوان ورودي و هم به عنوان خروجي عمل مي كنند.

CPU

Central Proccessing Unit

پردازش داده ها و كنترل بخش هاي مختلف سيستم را بر عهده دارد.
مهمترين ملاك در كارايي و سرعت سيستم است.



واحد سرعت CPU:

- MHz مگاهرتز
- GHz گيگا هرتز

شركت هاي توليد كننده CPU:

- Intel
- AMD

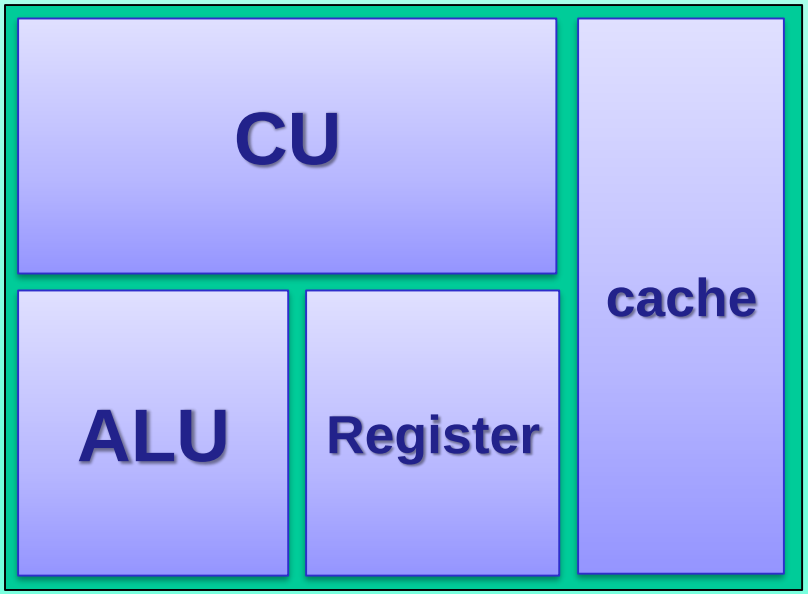
قسمت های مختلف CPU

۱- ALU (Arithmatic logic unit)

۲- CU (control unit)

۳- Register

۴- Cache (حافظه نهان)



CPU

قسمت های مختلف CPU

۱- **ALU** (Arithmatic logic unit): واحد محاسبه ومنطق که کلیه عملیات محاسبه و منطقی سیستم در این قسمت انجام می شود.

۲- **CU** (control unit): واحد کنترل که وظیفه انتخاب وتفسیر ونظارت بر اجرای دستورالعمل های برنامه یا فرمان در حال اجرا وهدایت و تداوم عملیات کل سیستم را بر عهده دارد.

۳- Register(ثبات): واحد های کوچک حافظه جهت نگهداری سریع وموقت نتایج در cpu به کار می رود.

۴- Cache حافظه نهان: حافظه ای که آخرین اطلاعات پردازش شده توسط CPUرا بطور موقت ذخیره نگه می دارد. هرچقدر حجم این حافظه بیشتر باشد سرعت پردازش داده ها بیشتر می شود.

فن خنک کننده CPU

برای پایین نگه داشتن دمای cpu لازمست از فن خنک کننده استفاده کنیم.

هنگام اسمبل کردن (یعنی مونتاژ قطعات کامپیوتر) می بایست فن خنک کننده را متناسب با قدرت و سرعت cpu انتخاب کنیم.

درصورت خرابی فن:

- کامپیوتر بخودی خود restart می کند.
- کامپیوتر بخودی خود shut down می کند.
- کامپیوتر متوقف میشود. (اصطلاحا قفل یا hang می کند)

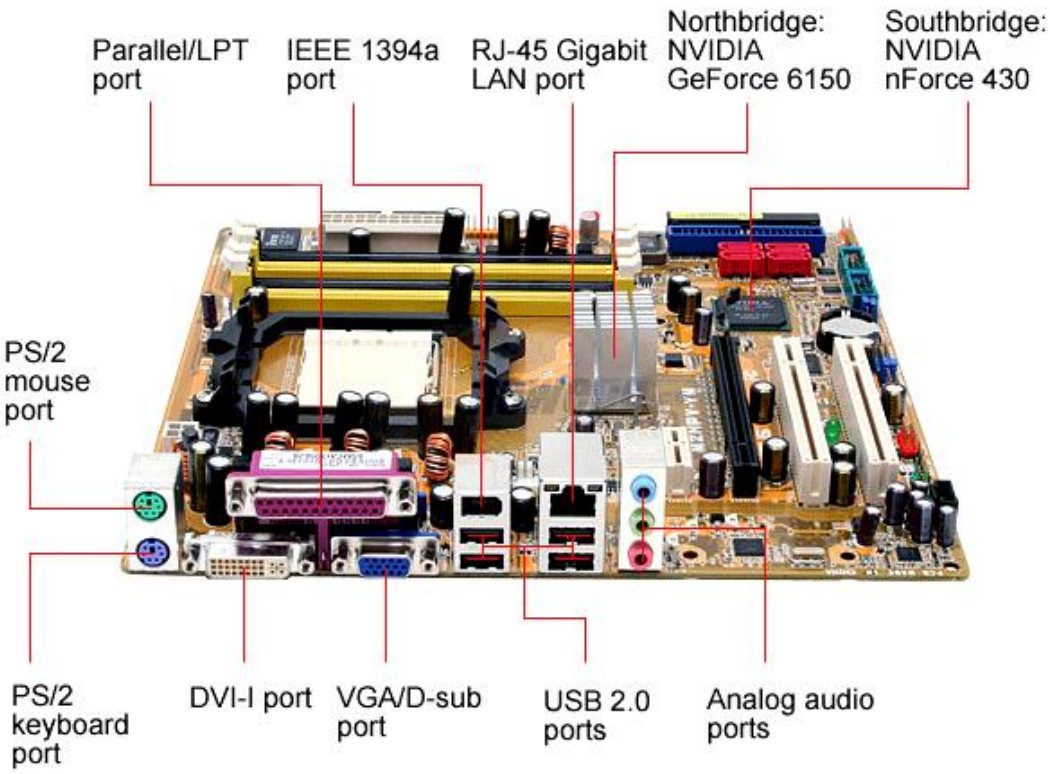
مادربرد Mother Board

اصلی ترین برد هر کامپیوتر که تمامی قطعات دیگر کامپیوتر بطور مسقیم و یا از طریق کابل با آن ارتباط دارند.

ارتباط دهنده بین قطعات سیستم می باشد.

مادربرد Mother Board

بعضی مادربردها خود دارای کارت گرافیک، صدا، مودم، کارت شبکه و ... هستند که در این صورت به این کارت ها Onboard گوییم.



- **عموما کارت هاي Onboard از امکانات و قدرت کمتري نسبت به موارد غیر Onboard برخوردارند.**

حافظه RAM

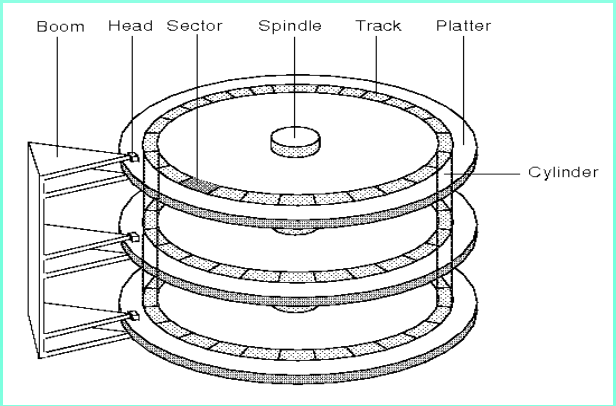
Random Access Memory

- محل ذخیره موقت اطلاعاتی است که CPU می خواهد روی آن پردازش انجام دهد.
- بطور مستقیم بر روی مادربرد نصب می شود.
- هر چه ظرفیت آن بالاتر باشد سرعت سیستم بالاتر می رود.

- اطلاعات Ram با قطع برق پاک می شوند.
- بدون Ram کامپیوتر روشن نخواهد شد به همین خاطر به آن حافظه اصلی گوییم.
- به هنگام خرید Ram می بایست دقت کنیم که حافظه خریداری شده با مادربرد سازگار باشد.

دیسک سخت H.D.D
(Hard Disk Drive)

- محل ذخیره دائم اطلاعات است.
- اطلاعات hard با قطع برق پاک نمی شوند.
- معمولا ظرفیت های بالایی دارند.
- نسبت به سایر دیسک ها سرعت بالاتری دارد.



محفظه DVD

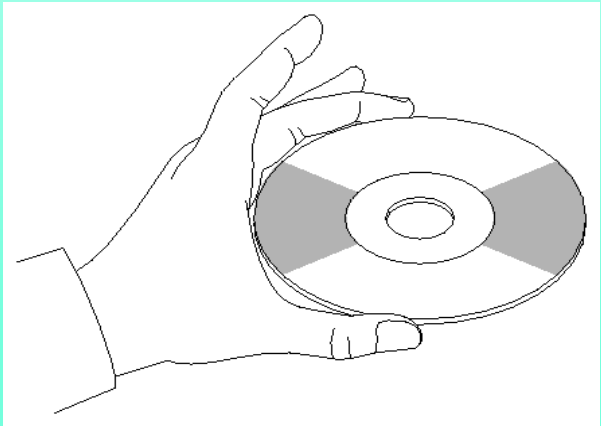
DVD Drive: محل قرارگیری CD یا DVD جهت خواندن اطلاعات.
معمولا ظرفیت 4.7GB دارند.

DVD Writer: محل قرارگیری CD یا DVD جهت خواندن و نیز ذخیره ونوشتن درآن.

DVD , CD

اطلاعات با تکنولوژی نور لیزر ذخیره می شوند.

نسبت به خش و ضربه آسیب پذیرند.



قالب های مختلف:

۱- data: جهت ذخیره هرنوع اطلاعات و نرم افزار ها

۲- Video CD: قالب مخصوص فیلم های ویدیویی

۳- Audio CD: فرمت ضبط سی دی های صوتی

کارت گرافیک VGA
Video Graphics Array

رابط بین اطلاعات متنی یا گرافیکی ارسال شده از CPU وصفحه نمایش (مانیتور) است.

هرچه کیفیت کارت گرافیک بالاتر باشد، تصاویر نمایش داده شده در مانیتور مطلوب تر است.

مودم Modem

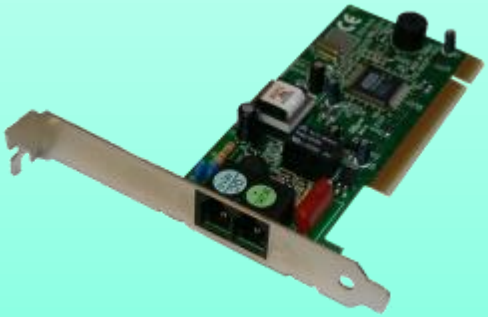
مودم از طریق سیم تلفن به خط تلفن ساختمان وصل می شود.

توسط مودم می توان تماس تلفنی و ارسال و دریافت فکس را نیز انجام داد.

مودم Modem

از طریق خط تلفن ارتباط کامپیوتر با اینترنت و دیگر شبکه ها را فراهم میکند.

Internal: به قطعاتي كه داخل case قرار بگيرند.



External: به قطعاتي كه خارج از case قرار بگيرند.

در 2نوع

کارت شبکه

برای اتصال به شبکه نیاز به کارت شبکه داریم.
کارت شبکه ارتباط بین یک سیستم و شبکه را برقرار میکند.

هرچه کیفیت این کارت بهتر باشد، سرعت کارکرد سیستم در شبکه بالاتر است.

کارت شبکه

اندازه و ابعاد پورت کارت شبکه بزرگتر از پورت مودم است.

منبع تغذیه برق سیستم Power

محل ورود برق به سیستم است و از آنجا به کلیه قطعات سیستم برق توزیع می شود.

انواع نمایشگر Monitor

CRT

Flatron

LCD

Plasma

تعريف عام حافظه :

- هر دستگاهي كه قادر به نگهداري اطلاعات باشد وبتوان اطلاعات را در آن ذخيره كرد به نحوي كه استفاده كننده از آن بتواند در هر لحظه كه لازم باشد به اطلاعات مورد نيازش دسترسي داشته باشد.

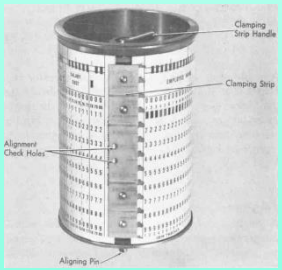
حافظه به دو دسته تقسيم ميشود :

- ***حافظه اوليه:*** **اصلی – درون ماشینی :** که پردازنده جهت اجرای برنامه مستقیما با ان سروکار دارند.
- ***حافظه ثانويه:*** **جانبی – برون ماشینی** : که جهت ضبط اطلاعات و فایلها به کار میرود .

✓ ميتوان گفت يك سيستم كامپيوتري از دو بخش برون ماشيني و درون ماشيني تشكيل شده است.

منظور از محیط برون ماشینی :

تجهیزات جانبی آن مثل **هارد دیسک، پرینتر** و غیره میباشد



انواع حافظه های برون ماشینی :

۱- کارت پانچ شدنی

۲- نوار پانچ شدنی

۳- انواع نوار مغناطیسی

۴- انواع دیسکهای مغناطیسی

۵- طبله Drum

۶- دیسکهای نوری

۷- دیسکهای نوری مغناطیسی

خصوصیات مشترک انواع حافظه :

1. نوشتن و خواندن
2. نشانی پذیری
3. قابلیت دستیابی(آدرس دهی)
4. ظرفیت
5. زمان دستیابی
6. نرخ انتقال یا سرعت انتقال rate
7. فرار و غیر فرار یا مانا و نامانا

خصوصيات مشترک انواع حافظه :

1- نوشتن و خواندن : هر حافظه اي قابليت دارد که بتوان بر آن نوشت (درج اطلاعات) و يا از آن خواند (واکشي اطلاعاتFetch) . مثلا در RAM و هارددیسك (H.D.D) هم ميتوان نوشت هم خواند . اما در حافظه اي مانند ROM و CD-ROM فقط ميتوان اطلاعات را خواند.

2- نشاني پذيري : هر حافظه اي مجهز است به يك مکانيسم نشاني پذيري.

3- قابليت دستيابي(آدرس دهي) : هر حافظه اي از طريق مکانيسم نشاني دهي مورد دستيابي قرار ميگيرد . هر حافظه اي داراي يك شيوه آدرس دهي ميباشد که به کمك ان خانه هايش قابل دستيابي ميگردد مثلا حافظه RAMآرايه اي از بايتهاست که هر کدام يك آدرس (عدد يکتا) دارند . يا هارد ديسك به صورت سه جفت عدد (شماره هد / شماره سيلندر / شماره سکتور) آدرس دهي ميشود.

4- ظرفيت : هر حافظه اي داراي ظرفيتي است که به بيت يا بايت و... بيان مي شود.

خصوصيات مشترک انواع حافظه :

5- زمان دستيابي : مدت زماني است که براي خواندن و نوشتن به کار ميرود. از لحظه اي که دستور خواندن و نوشتن داده ميشود تا هنگامي که حافظه مورد دستيابي قرار ميگيرد را زمان دستيابي گويند.

- مثلا زمان دستيابي حافظه هاي RAMدر حدود 120 نانوثانيه است و زمان دستيابي ديسک حدود 30 ميلي ثانيه است . يعني RAMحدود 250 برابر سريعتر از ديسک است

6- نرخ انتقال يا سرعت انتقال rate : کميتي از اطلاعات که در واحد زمان از حافظه قابل انتقال است واحد ان بايت در ثانيه است .

- در زير ليستي از حافظه ها براساس کاهش سرعت و افزايش ظرفيت را ميبينيد :

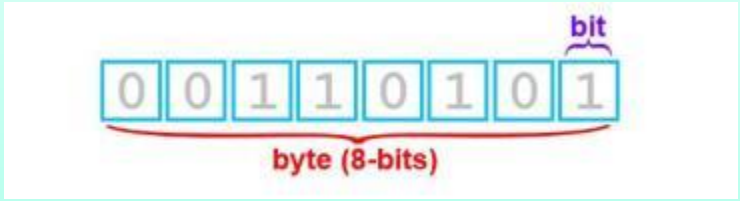
1. ثبات 2. حافظه نهان cacheکش 3. حافظه اصلي 4. حافظه Flash 5. ديسک مغناطيسي 6. ديسک نوري 7. نوار مغناطيسي

7- فرار و غير فرار يا مانا و نامانا : حافظه هايي که با رفتن برق اطلاعات انها از بين ميرود حافظه هاي فرار volatileو آنهايي که با رفتن برق داده هاي خود را حفظ ميکنند غيرفرار non volatileميگويند . حافظه هاي اصلي اغلب فرار و حافظه هاي جانبي غير فرار هستند

واحد های حافظه

- اساس کار کامپیوترها بصورت دیجیتال و دودویی (باینری) است.
- کامپیوتر همه اطلاعات را بصورت دودویی ذخیره می کند.
- اطلاعات بصورت دودویی پردازش میشوند.

واحدهای حافظه:



- کوچکترین واحد حافظه BIT نام دارد.
- به هر هشت بیت یک بایت (Byte) گوییم

بیت BIT: بیت کوچک‌ترین واحد حافظه است که فقط دو مقدار صفر (۰) یا یک (۱) را می‌توان در آن ذخیره کرد.

بایت Byte: هر بایت برابر ۸ بیت است، معمولاً حجم هر کارکتری (کاراکتر یعنی ارقام، حروف یا علامتها) برابر یک بایت است، به عبارتی هر کاراکتر یک بایت فضا اشغال می‌کند.

کیلوبایت KB: هر کیلوبایت برابر ۱۰۲۴ بایت است، به عبارتی هر کیلوبایت برابر 2^{10} بایت است.

مگابایت MB: هرمگابایت برابر ۱۰۲۴ کیلوبایت است، به عبارتی هرمگابایت برابر 2^{10} کیلوبایت است.

گیگابایت GB: هر گیگابایت برابر ۱۰۲۴ مگا بایت است، به عبارتی هر گیگابایت برابر 2^{10} مگابایت است.

ترابایت TB: هر ترابایت برابر ۱۰۲۴ گیگابایت است، به عبارتی هر ترابایت برابر 2^{10} گیگابایت است.

پتابایت PB: هر پتابایت برابر ۱۰۲۴ ترابایت است، به عبارتی هر پتابایت برابر 2^{10} ترابایت است.

اگزابایت EB: ر اگزابایت برابر ۱۰۲۴ پتابایت است، به عبارتی هر اگزابایت برابر 2^{10} پتابایت است.

نرم افزار Software

به برنامه های قابل اجرا در کامپیوتر به منظور انجام هدفی خاص.

برنامه ها توسط برنامه نویسان تولید میشود.

به هر نسخه از نرم افزار Version گوییم.

نرم افزار Software

انواع نرم افزار:

۱- نرم افزارهای سیستمی (System Softwares):

ارتباط دهنده کاربر یا سایر نرم افزارها با کامپیوتر است.

۲- نرم افزارهاي کاربردی (Application Softwares):

مجموعه نرم‌افزارهایی هستند که برای تامین نیازهای متنوع کاربران تولید می شوند.

نرم افزار Software

۱- انواع نرم افزارهای سیستمی (System Softwares):

الف)سیستم عامل: اداره کننده سخت افزار کامپیوتر است که نرم افزارهاي ديگر بدون آن قادر به اجرا نخواهند بود. **مانند: Dos, Linux, Windows, Mac ,...**

ب)برنامه‌هاي كمكي: مانند برنامه هاي پاكسازي كننده ويروس :Antivirus
McAfee, Norton, Nod32, CasperSky ,...

ج)مترجم ها و مفسرها: تبديل برنامه نوشته شده كاربر، به زبان صفرو يك كامپيوتر.
Programming:
Assembly, Java, Pascal, C, C++, C#.Net, Visual Basic.Net

نرم افزار Software

2- انواع نرم افزارهاي كاربردي (Application Software):

نرم افزارهاي فشرده ساز:

WinZip, WinRar

نرم افزارهاي صوتي و تصويري:

Multimedia:

JetAudio, Windows Media Player, WinAmp, PowerDVD

نرم افزارهاي گرافيكي:

Graphics:

Photoshop, Corel, Freehand, Paint

نرم افزار Software

2- انواع نرم افزارهاي كاربردي (Application Software): (ادامه)

نرم افزارهاي انيميشن ساز:

Animation:

3Dmax – Flash – Swish – Maya

نرم افزارهاي مهندسي:

Engineeing:

Autocad – Protel – Pspice - Catia - Matlab

نرم افزارهاي فيلم سازي يا تدوين:

Movie Maker:

Adobe Premiere, Ulead Video Studio, Windows Movie Maker

نرم افزار Software

2- انواع نرم افزارهاي كاربردي (Application Software): (ادامه)

نرم افزارهاي كاوشگر سايت هاي اينترنتي:

Internet Browser:

Internet Explorer, Firefox, Opera, Chrome, Safari (Mac)

نرم افزارهاي صفحه آرايي:

Page Maker:

Adobe Indesign, Quark, Office

مبناهای عددی یا سیستم اعداد

- ✓مبنای ۱۰ Decimal
- ✓مبنای ۲ Binary
- ✓مبنای ۸ Octal
- ✓مبنای ۱۶ Hexa

سیستم اعداد دهدهی (Decimal)

سیستم اعداد دهدهی یکی از سیستم های متداول است که همگان روزانه با آن سر وکار دارند. در این سیستم هر عدد می تواند ترکیبی از اعداد **۰ تا ۹** باشد . هر عدد در سیستم دهدهی را می توان به صورت زیر نمایش داد:

$$a_k\,(10)^k$$

Decimal
0
1
2
3
4
5
6
7
8
9

مثال / اثبات:

$$(\overset{2}{2}\overset{1}{3}\overset{0}{4})_{10} = ?$$

$$\underset{100}{2*10^2} + \underset{10}{3*10^1} + \underset{1}{4*10^0} = 234$$

2*100 + 3*10 + 4*1 = 234

سیستم اعداد دودویی (Binary)

در این سیستم، مبنای اعداد ، ۲ است، لذا هر عدد در این سیستم می تواند ترکیبی از ارقام **۰ و ۱** باشد ، مانند ۱۱۱ و ۱۰۱۱ و ۱۰۰۰۱۱۱۱ . شکل کلی اعداد این سیستم به صورت زیر است:

$$a_k (2)^k$$

Decimal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

نکته۱: از نظر تئوری می توان سیستم های عددی زیادی را تعریف کرد . ولی سیستم های مطرح شده ، از اهمیت ویژه ای برخوردارند و ما در با آنها بیشتر سر و کار داریم.

نکته۲: مبنای کار کامپیوتر بصورت دودویی (Binary) می باشد. تمامی دستورات در کامپیوتر به صفر و یک تبدیل شده و سپس توسط CPU اجرا می شود.

سیستم اعداد هشتایی (Octal)

در این سیستم ، مبنای اعداد ، ۸ است ، لذا هر عدد در این سیستم می تواند ترکیبی از ارقام **۰ تا ۷** باشد . نمایش عدد در این سیستم به صورت زیر است:

$$a_k\ (8)^k$$

Decimal	Binary	Octal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

سیستم اعداد شانزده تایی (Hexa)

Decimal	Binary	Octal	Hexa
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	---	8
9	1001	---	9
10	1010	---	A
11	1011	---	B
12	1100	---	C
13	1101	---	D
14	1110	---	E
15	1111	---	F

در این سیستم، مبنای اعداد ۱۶ است، لذا می توان از ۱۶ رقم در نوشتن اعداد این سیستم استفاده کرد. چون اعداد ۹ به بالا را به عنوان یک رقم نمی شناسیم، برای نمایش ۱۰ تا ۱۵ از علائم **A تا F** استفاده می کنیم . شکل کلی نمایش این اعداد به صورت زیر است:

$$a_k (16)^k$$

قانون کلی تبدیل اعداد

❑ دهدهی به مبناهای دیگر

بطور کلی برای تبدیل یک عدد در مبنای ده به هر مبنای دیگر می بایست عدد دهدهی را مرتبا تقسیم برآن مبنا کنیم. این عمل را آنقدر انجام می دهیم تا خارج قسمت صفر شود. سپس باقیمانده ها را از راست به چپ می نویسیم.

❑ مبنای دلخواه به مبنای ده

به طور کلی برای تبدیل یک عدد در مبنای دلخواه به مبنای ده، به هر رقم یک موقعیت می دهیم. موقعیت ها از سمت راست به چپ و از صفر شماره گذاری می شوند. سپس هر رقم را ضرب در مبنا به توان موقعیت کرده و در آخر اعداد حاصله را با هم جمع می کنیم.

چند نکته برای اعداد زوج و فرد

1. عدد فرد در هر مبنایی فرد است و عدد زوج در هر مبنایی زوج است.
2. در مبنای **شانزده** اگر سمت راست عدد ارقام (A,C,E) باشند آن عدد زوج است.
3. در مبنای **شانزده** اگر سمت راست عدد ارقام (B,D,F) باشند آن عدد فرد است.
4. در مبنای **دو** اگر سمت راست صفر باشد عدد زوج است و اگر یک باشد عدد فرد است.

تبدیل اعداد دهدهی صحیح به دودویی و بالعکس

- اعداد دهدهی زیر را به مبنای دو ببرید؟

مثال:

روش تقسیم متوالی

$(25)_{10} = (11001)_2$

$(32)_{10} = (100000)_2$

$(1001)_{10} = (1111101001)_2$

تبدیل اعداد دهدهی صحیح به دودویی و بالعکس

❑ اعداد دودویی زیر را به مبنای ده ببرید؟

مثال:

$$\overset{4\ 3\ 2\ 1\ 0}{(11001)_2} = (25)_{10}$$

$1*2^0 + 0*2^1 + 0*2^2 + 1*2^3 + 1*2^4 \rightarrow 1 + 0 + 0 + 8 + 16 = 25$

$$\overset{5\ 4\ 3\ 2\ 1\ 0}{(100010)_2} = (34)_{10}$$

$0*2^0 + 1*2^1 + 0*2^2 + 0*2^3 + 0*2^4 + 1*2^5 \rightarrow 0 + 2 + 0 + 8 + 32 = 34$

$$\overset{6\ 5\ 4\ 3\ 2\ 1\ 0}{(1100101)_2} = (101)_{10}$$

$1*2^0 + 0*2^1 + 1*2^2 + 0*2^3 + 0*2^4 + 1*2^5 + 1*2^6 \rightarrow$

$1 + 0 + 4 + 0 + 0 + 32 + 64 = 101$

تبدیل اعداد دهدهی صحیح به هشت تایی و بالعکس

- اعداد دهدهی زیر را به مبنای هشت ببرید؟

مثال:

روش تقسیم متوالی

$(25)_{10} = (3\ 1)_8$

$(32)_{10} = (4\ 0)_8$

$(1001)_{10} = (1\ 7\ 1\ 1)_8$

تبدیل اعداد دهدهی صحیح به هشت تایی و بالعکس

- اعداد هشتایی زیر را به مبنای ده ببرید؟

مثال:

$(31)_8 = (25)_{10}$

$(40)_8 = (32)_{10}$

$(1711)_8 = (969)_{10}$

تبدیل اعداد دهدهی صحیح به شانزده تایی و بالعکس

- اعداد دهدهی زیر را به مبنای شانزده ببرید؟

مثال:

روش تقسیم متوالی

$(25)_{10} = (19)_{16}$

$(32)_{10} = (20)_{16}$

$(1001)_{10} = (3E9)_{16}$

تبدیل اعداد دهدهی صحیح به شانزده تایی و بالعکس

- اعداد شانزده تایی زیر را به مبنای ده ببرید؟

مثال:

$(19)_{16} = (25)_{10}$

$(20)_{16} = (32)_{10}$

$(3E9)_{16} = (1001)_{10}$

تبدیل اعداد دودویی به هشت تایی و بالعکس

❑ دودویی به هشت تایی

روش اول:

ابتدا روش بسط سپس روش تقسیم متوالی

$(1110001)_2 \rightarrow (113)_{10}$ روش بسط

$(113)_{10} \rightarrow (161)_8$ روش تقسیم متوالی

روش دوم:

سه رقم سه رقم از سمت راست جدا میکنیم

$(001\ 110\ 001)_2 \rightarrow (161)_8$ طبق جدول

1 6 1

❑ هشت تایی به دودویی

روش اول:

بردن عدد به مبنای دو و جدا کردن چهار رقم چهار رقم از سمت راست

روش دوم:

ابتدا روش بسط (مبنای ۱۰) سپس روش تقسیم متوالی (مبنای ۱۶)

$(161)_8 \rightarrow (1110001)_2$

تبدیل اعداد دودویی به شانزده تایی و بالعکس

❑ دودویی به شانزده تایی

روش اول:

ابتدا روش بسط سپس روش تقسیم متوالی

$(1111101)_2 \rightarrow (125)_{10}$ روش بسط

$(125)_{10} \rightarrow (7D)_{16}$ روش تقسیم متوالی

روش دوم:

چهار رقم چهار رقم از سمت راست جدا میکنیم

$(\underset{7}{0111}\ \underset{D}{1101})_2 \rightarrow (7D)_{16}$ طبق جدول

❑ شانزده تایی به دودویی

روش اول:

بردن عدد به مبنای دو و جدا کردن چهار رقم چهار رقم از سمت راست

روش دوم:

ابتدا روش بسط (مبنای ۱۰) سپس روش تقسیم متوالی (مبنای ۱۶)

$(7D)_{16} \rightarrow (1111101)_2$

تبدیل اعداد شانزده تایی به هشت تایی و بالعکس

❑ شانزده تایی به هشت تایی

روش اول:
برای اینکار ساده تر آن است که ابتدا عدد را به مبنای ۲ برده و سپس به مبنای هشت دیگر ببریم (سه رقم سه رقم)

$(A36)_{16}$ ➔ $(1010\ \ 0011\ \ 0110)_2$ طبق جدول

$(\underset{5}{101}\ \underset{0}{000}\ \underset{6}{110}\ \underset{6}{110})_2$ ➔ $(5066)_8$

روش دوم:
ابتدا روش بسط (مبنای ۱۰) سپس روش تقسیم متوالی (مبنای ۸)

$(A36)_{16}$ ➔ $(2614)_{10}$ روش بسط

$(2614)_{10}$ ➔ $(5066)_8$ روش تقسیم متوالی

❑ هشت تایی به شانزده تایی

روش اول:
بردن عدد به مبنای دو و جدا کردن چهار رقم چهار رقم از سمت راست

روش دوم:
ابتدا روش بسط (مبنای ۱۰) سپس روش تقسیم متوالی (مبنای ۱۶)

$(5066)_8$ ➔ $(A36)_{16}$

جمع اعداد دودویی

مثال در مورد اعداد دهدهی

۲۵۰	
۲۵۰	+

۵۰۰	

جمع بیتی اعداد باینری

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 1 0
1 + 1 + 1 = 1 1

جمع اعداد دودویی (ادامه)

مثال:

	۵		۱۰۱	
+	۷		۱۱۱	+
	ـــ		ـــــــ	
	۱۲		۱۱۰۰	

	۱۱۱۱۱۰۱۰		۲۵۰	
+	۱۱۱۱۱۰۱۰	+	۲۵۰	+
	ـــ		ـــ	
	۱۱۱۱۱۰۱۰۰		۵۰۰	

	۱۴۷		۱۰۰۱۰۰۱۱	
+	۲۰۴		۱۱۰۰۱۱۰۰	+
	ـــ		ـــــــــ	
	۳۵۱		۱۰۱۰۱۱۱۱۱	

تمرین:

	۱۱۱۱۱۱۱۱۱۱		۱۰۰۱۰۰۱۰۰۱		۱۱۱۱۰۰۰۰۱۱۱۱		
+	۱۰۰۰۰۰۰۰۱۱	+	۱۱۱۱۱۱۱۱۱	+	۱۰۱۰۱۰۱۰۱۰۱۰	+	

تفریق اعداد دودویی

الگوریتم و فلوچارت

تعریف الگوریتم

- هر دستورالعملی که مراحل انجام کاری را با **زبانی دقیق** و با **جزئیات کافی** بیان نماید بطوریکه **ترتیب مراحل** و **شرط خاتمه عملیات** در آن کاملاً مشخص شده باشد را الگوریتم گویند.

ادامه تعریف الگوریتم

- منظور از **زبان دقیق،** آن است که الگوریتم دقیقا" به همان صورتیکه مورد نظر نویسنده است اجرا گردد.
- منظور از **جزئیات کافی،** آن است که در طول اجرای الگوریتم عملیات ناشناخته پیش نیامده و باعث انحراف از مسیر و هدف اصلی نگردد.
- منظور از **ترتیب مراحل،** آن است که مراحل اجرای الگوریتم قدم به قدم و با رعایت تقدم و تأ خر مشخص شده باشد.
- منظور از **شرط خاتمه،** پایان پذیر بودن الگوریتم می باشد و بهر حال الگوریتم باید در زمانی دلخواه و تحت شرایط یا شرایط داده شده خاتمه پذیرد.

مراحل تهیه الگوریتم

برای تهیه یک الگوریتم خوب و کارآمد باید مراحل خاصی اجرا شوند:

۱- تعریف دقیق مسئله: باید مسئله را تجزیه و تحلیل کرده تا کوچکترین ابهامی در فهم آن وجود نداشته باشد.

۲- تعیین عوامل اصلی(متغیرهای) مورد نیاز

۳- تعیین ورودی و خروجی مسئله : (داده ها و اطلاعات)

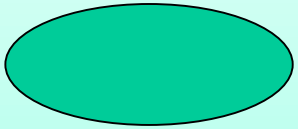
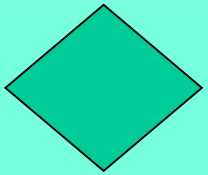
۴- بررسی راه حل های مختلف مسئله

۵- انتخاب یک راه حل مناسب

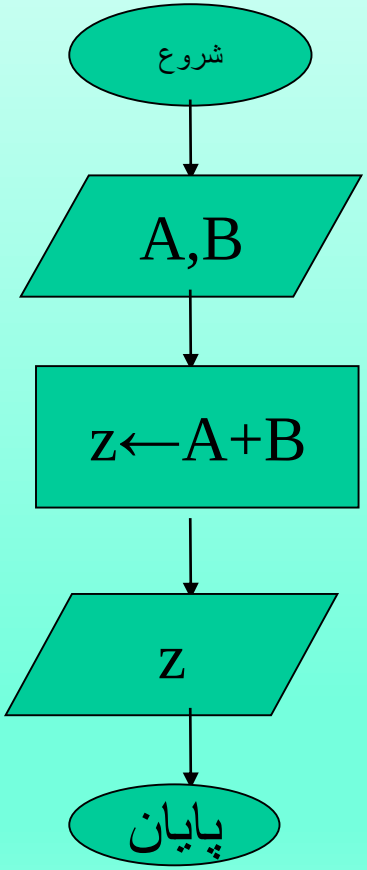
۶- اشکال زدایی

فلوچارت

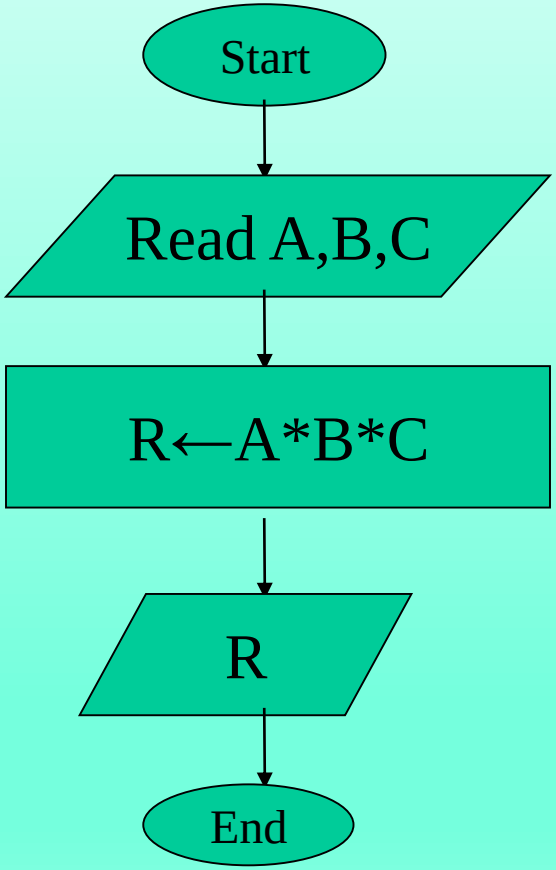
- بیان تصویری الگوریتم
- مراحل انجام کار با اشکال هندسی نشان داده می شوند.
- مراحل انجام کار توسط خطوط به هم وصل می گردند.

شکل	شرح	مثال
	برای نشان دادن شروع و خاتمه عملیات	start End
	محاسبات و مقداردهی	c←a+b d← i
	ورود اطلاعات خروج بر روی صفحه نمایش	A,B
	خروج اطلاعات بر روی کاغذ	A,B,”100”
	سئوال، تصمیم گیری و شرط های دلخواه	ورودی ? خروجی خروجی خروجی

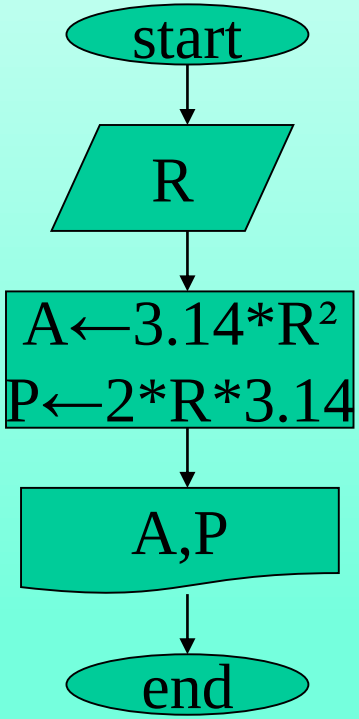
مثال ۱: فلوچارتی رسم کنید که دو عدد را خوانده و حاصل جمع آنها را نمایش دهد.



مثال ۱: الگوریتم و فلوچارتی برنامه ای را بنویسید که سه را خوانده و حاصلضرب آنها را نمایش دهد.



مثال۲: فلوچارتی رسم کنید که شعاع یک دایره را خوانده، مساحت و محیط آنرا چاپ نماید.

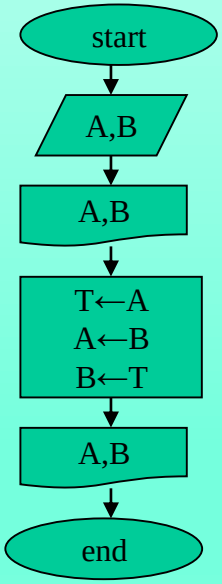


محیط: $P= 2R\pi$

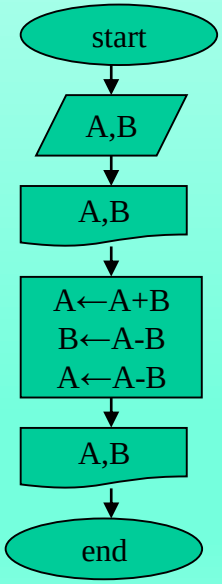
مساحت: $A= R^2 \pi$

مثال ۴: فلوچارت برنامه ای را رسم کنید که دو عدد را خوانده سپس مقادیر آن دو را با هم جابجا نماید.

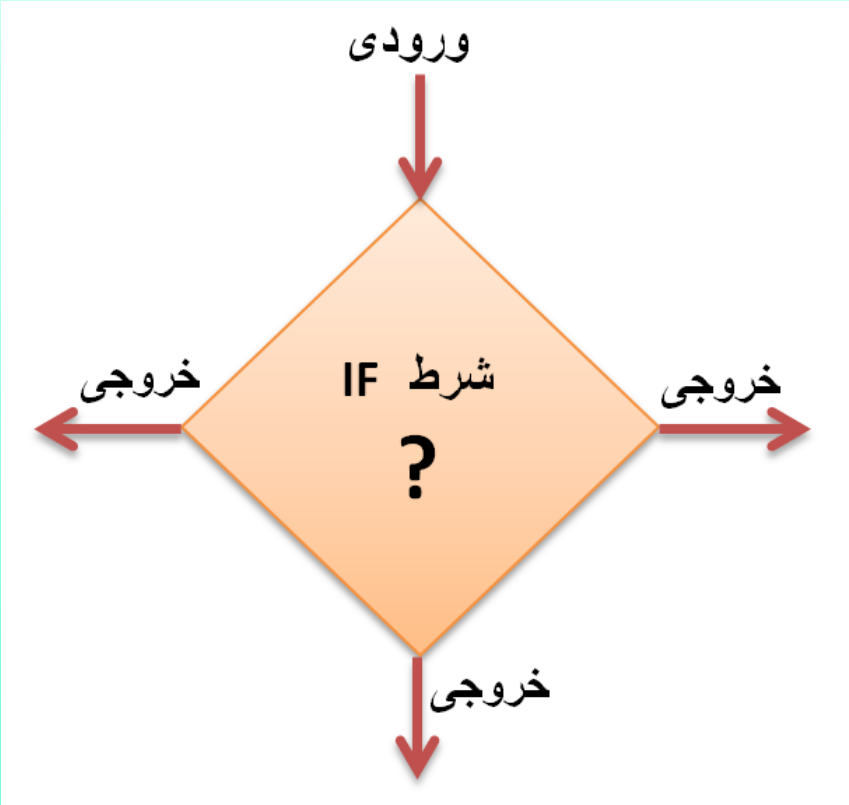
روش اول: استفاده ازمتغیر کمکی



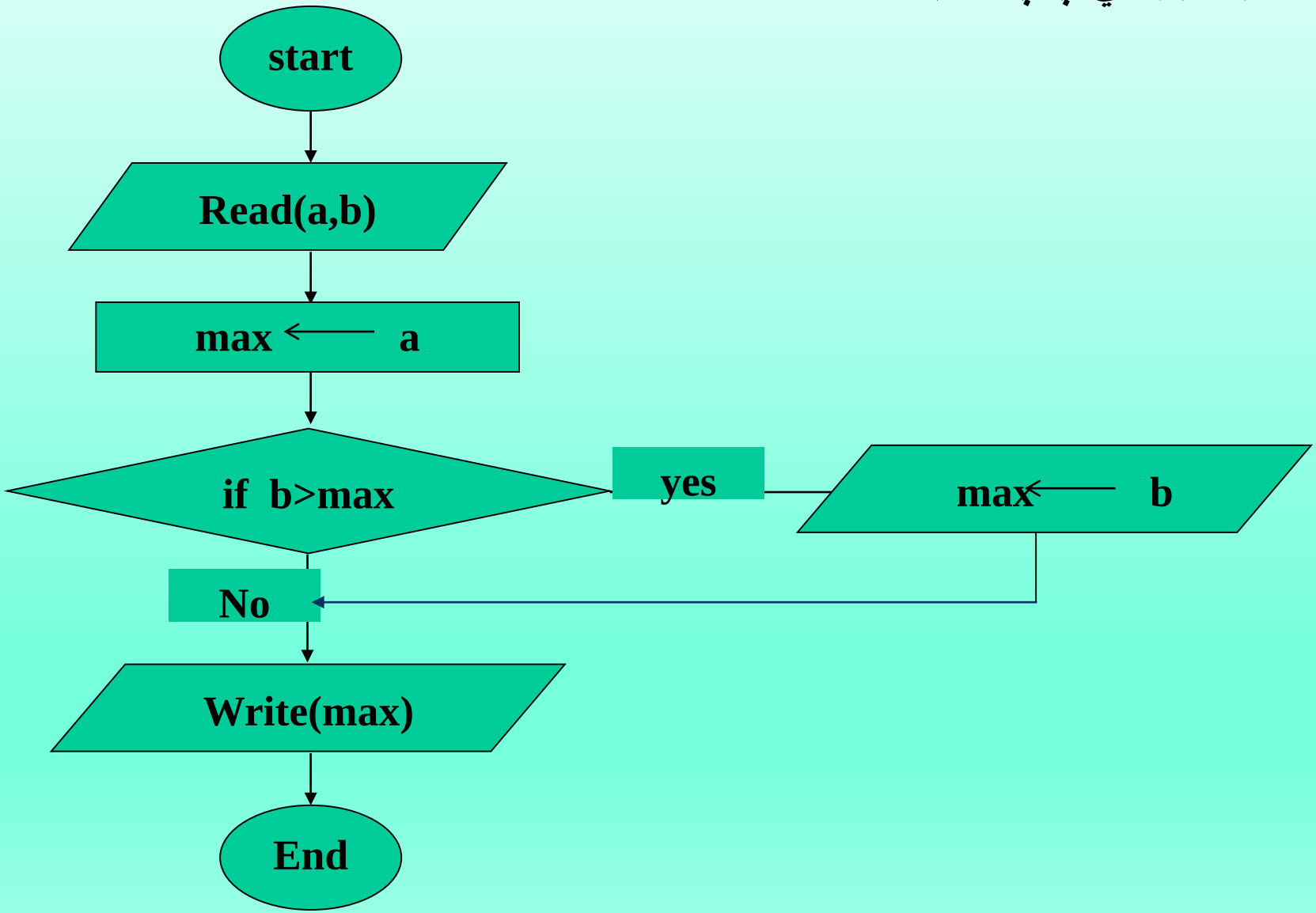
روش دوم: استفاده ازعملیات ریاضی



عبارات شرطی:

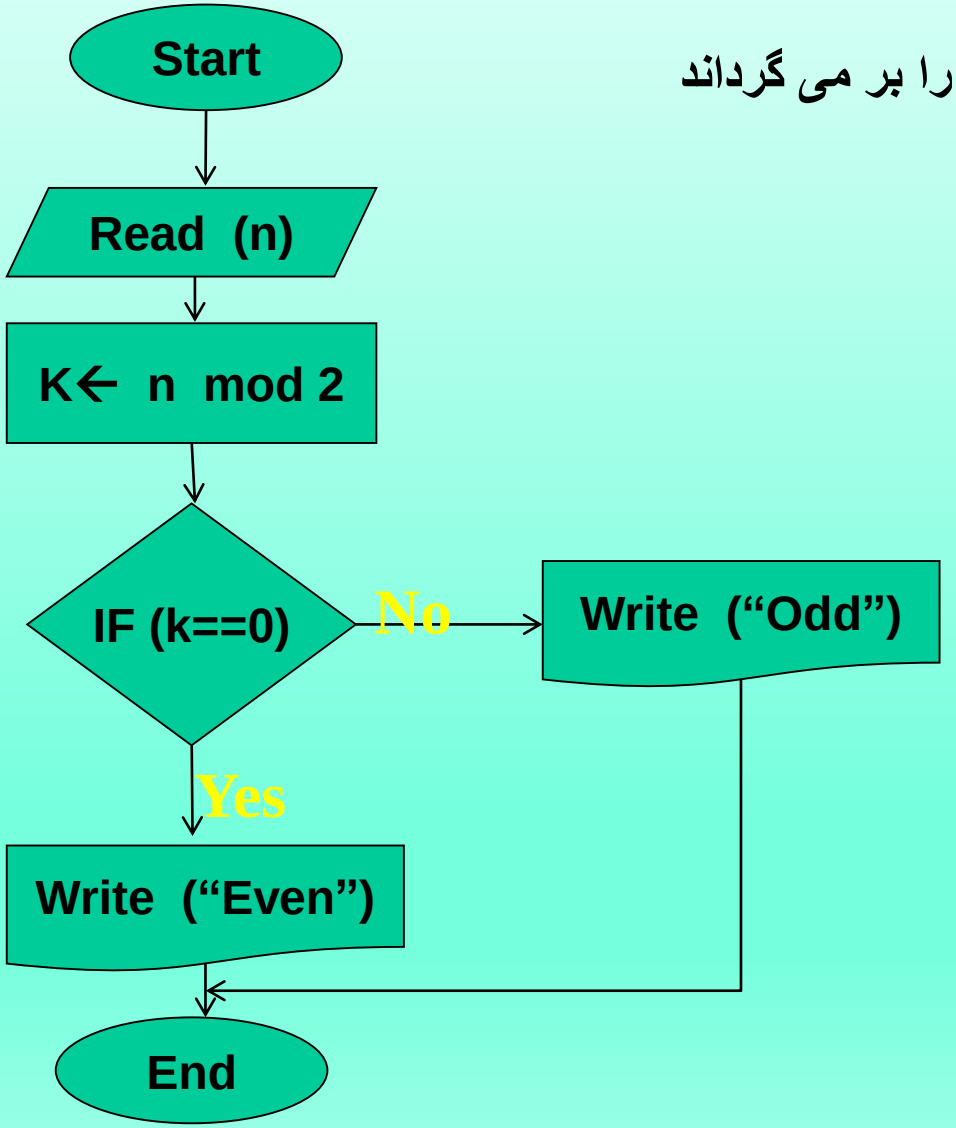


مثال : فلوچارتي رسم كنيد كه دو عدد از ورودي دريافت كرده بزرگترين عدد را پيدا كرده در خروجي چاپ نمايد.



مثال۵: فلوچارتی رسم نمائید که یک عدد از ورودی دریافت کرده فرد یا زوج بودن آنرا تشخیص دهد.

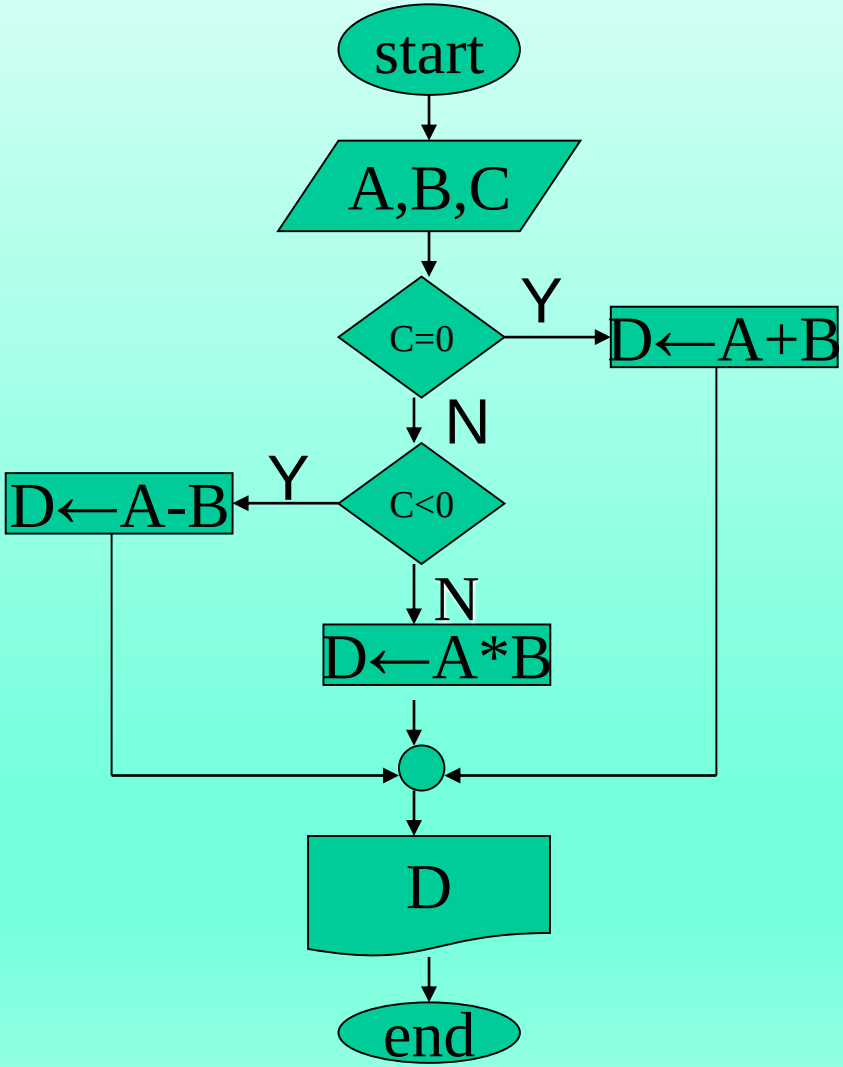
Mod: بعد از تقسیم باقی مانده را بر می گرداند



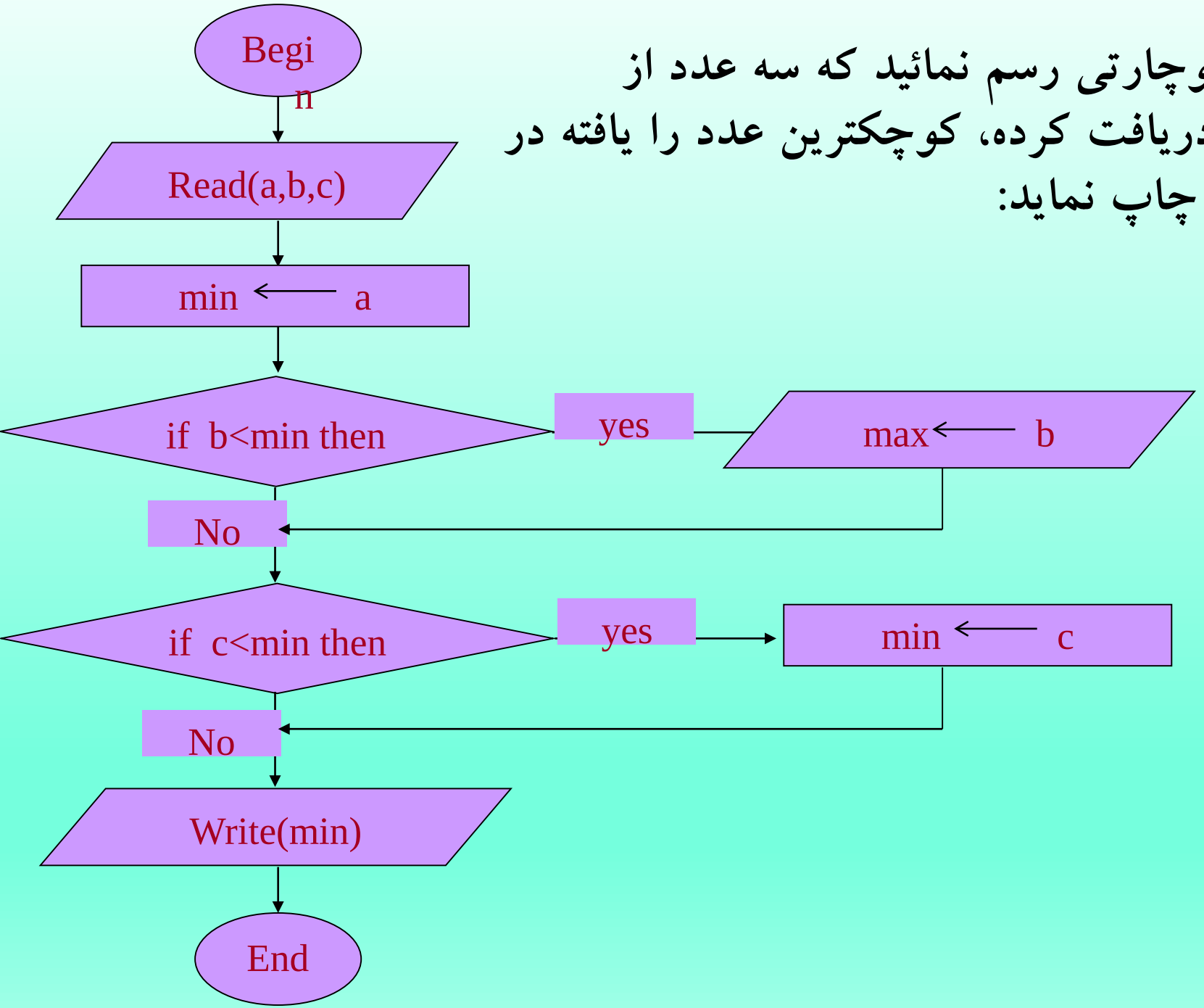
مثال۳: فلوچارتی رسم کنید که سه عدد را خوانده و بصورت زیر تصمیم گیری نماید را نمایش دهد.

الف: اگر عدد سوم صفر بود حاصل جمع دو عدد دیگر
ب: اگر عدد سوم منفی بود تفاضل دو عدد دیگر
ج: اگر عدد سوم مثبت بود حاصل ضرب دو عدد دیگر

حل مثال۳



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 می‌باشد:

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

الگوریتم های حلقوی

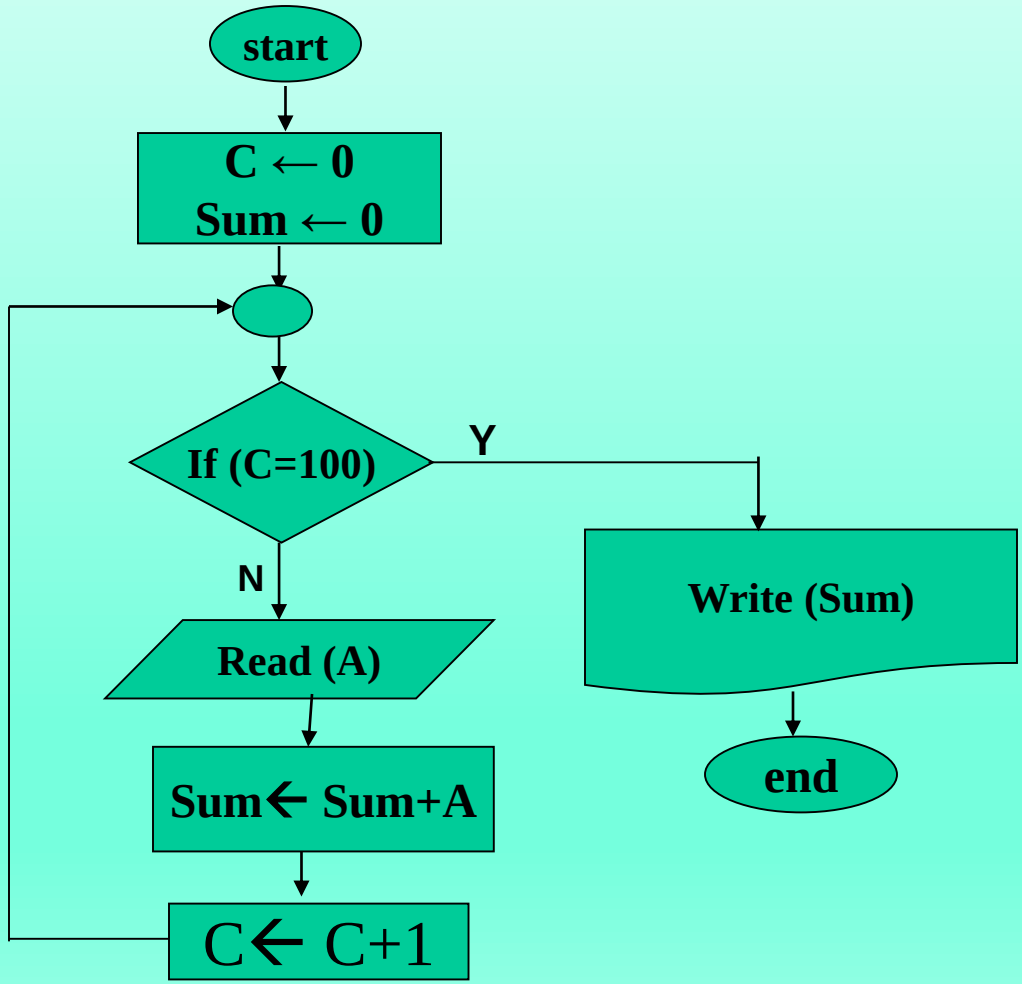
- **تعریف اول:** مراحلی از الگوریتم که چندین بار اجرای آنها تکرار می گردد تشکیل یک حلقه (LOOP) را می دهند.

- **الف: برای ساختن یک حلقه از یک متغیر کمکی استفاده می گردد، این متغیر را قبل از شروع حلقه با یک مقدار اولیه آماده می سازیم و سپس معمولاً در انتهای حلقه و قبل از بازگشت به ابتدای حلقه مقداری را به آن اضافه کرده و تحت یک شرایط خاص به مراحل قبل پرش می نماییم.**

- **ب: مقداری که قبل از شروع حلقه به متغیر حلقه داده می شود را مقدار اولیه یا شرط اولیه گویند.**

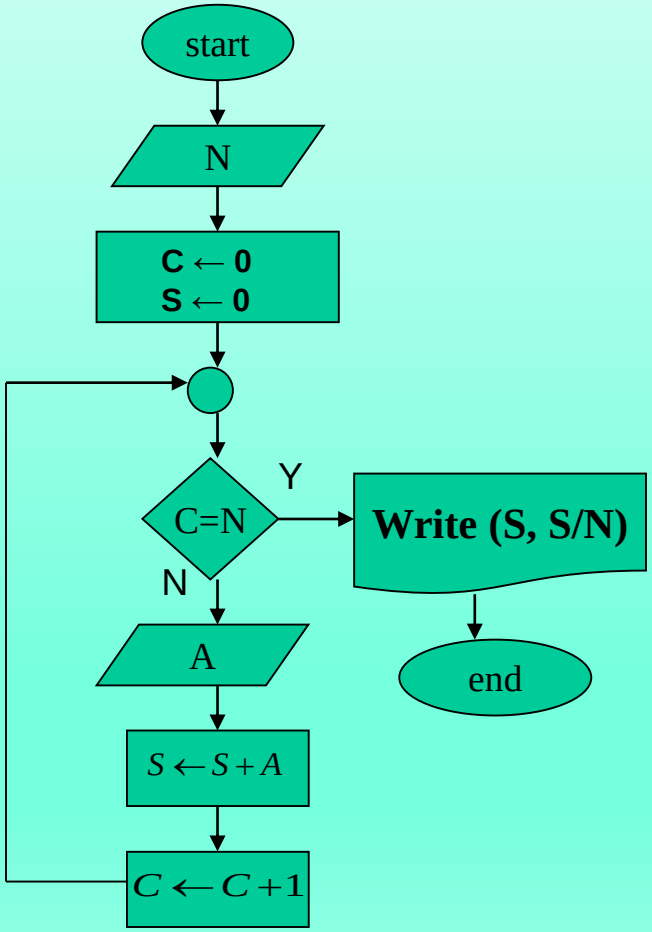
- **ج: مقداری که پس از یکبار اجرای مراحل حلقه به متغیر حلقه اضافه می شود را مقدار اضافه شونده می نامند.**

مثال : الگوریتم و فلوچارت برنامه ای را بنویسید که ۱۰۰ عدد از ورودی خوانده مجموع آنها را محاسبه و چاپ دهد.



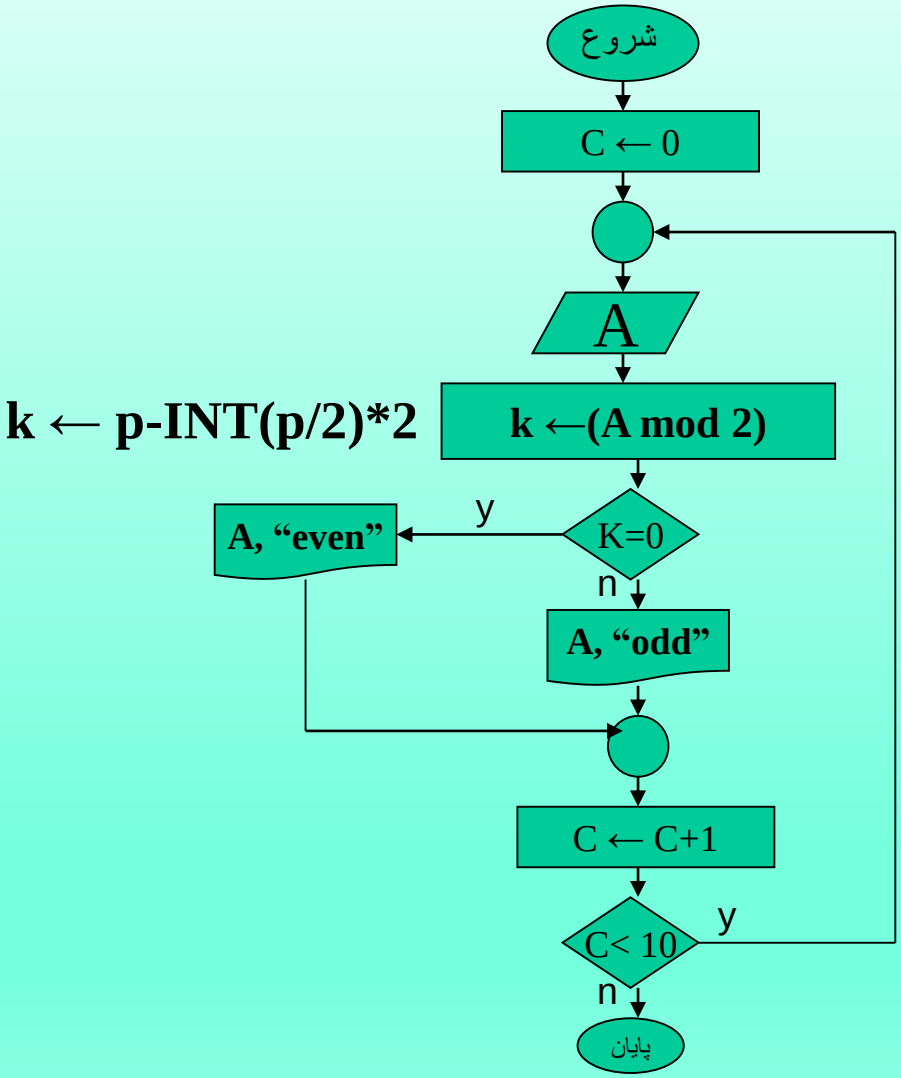
- Count شمارنده
- Sum مجموع
- A اعداد ورودی

مثال ۵: فلوچارتی رسم کنید که یک عدد بزرگتر از صفر را خوانده سپس به تعداد آن عدد, اعداد دیگری را خوانده مجموع و میانگین آنها را نمایش دهد.



- N عدد خوانده شده
- C شمارنده
- S مجموع
- A اعداد ورودی

مثال ۶: الگوریتم برنامه ای را بنویسید که ۱۰ عدد را گرفته و تعیین کند کدام زوج و کدام فرد است.



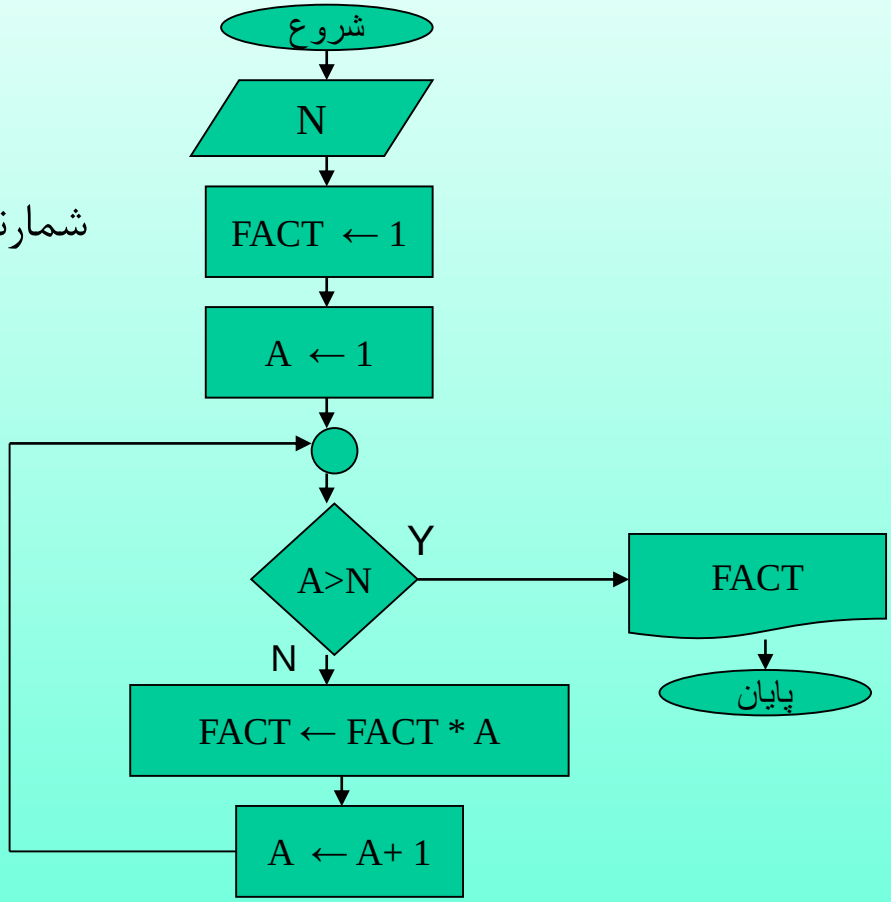
- C شمارنده
- A اعداد ورودی
- Even = زوج
- Odd = فرد

مثال ۱۰: الگوریتم برنامه ای را بنویسید که یک عدد مثبت را خوانده فاکتوریل آن را نمایش دهد.

- فاکتوریل یک عدد یعنی حاصل ضرب اعداد ۱ تا آن عدد
- فاکتوریل عدد صفر برابر یک می باشد.

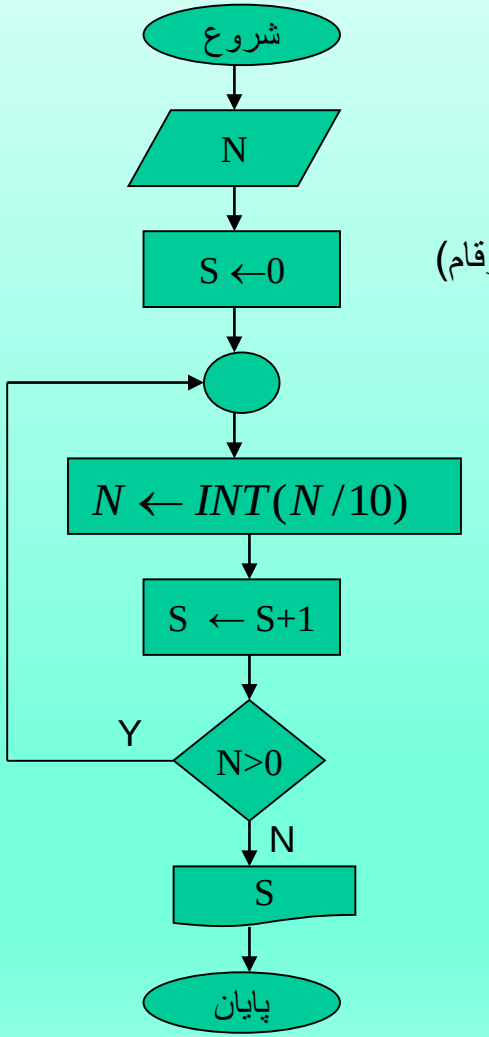
حل مثال ۱۰

- شمارنده A



مثال ۷: الگوریتم برنامه ای را بنویسید که یک عدد مثبت را خوانده و تعداد ارقام عدد را نشان دهد.

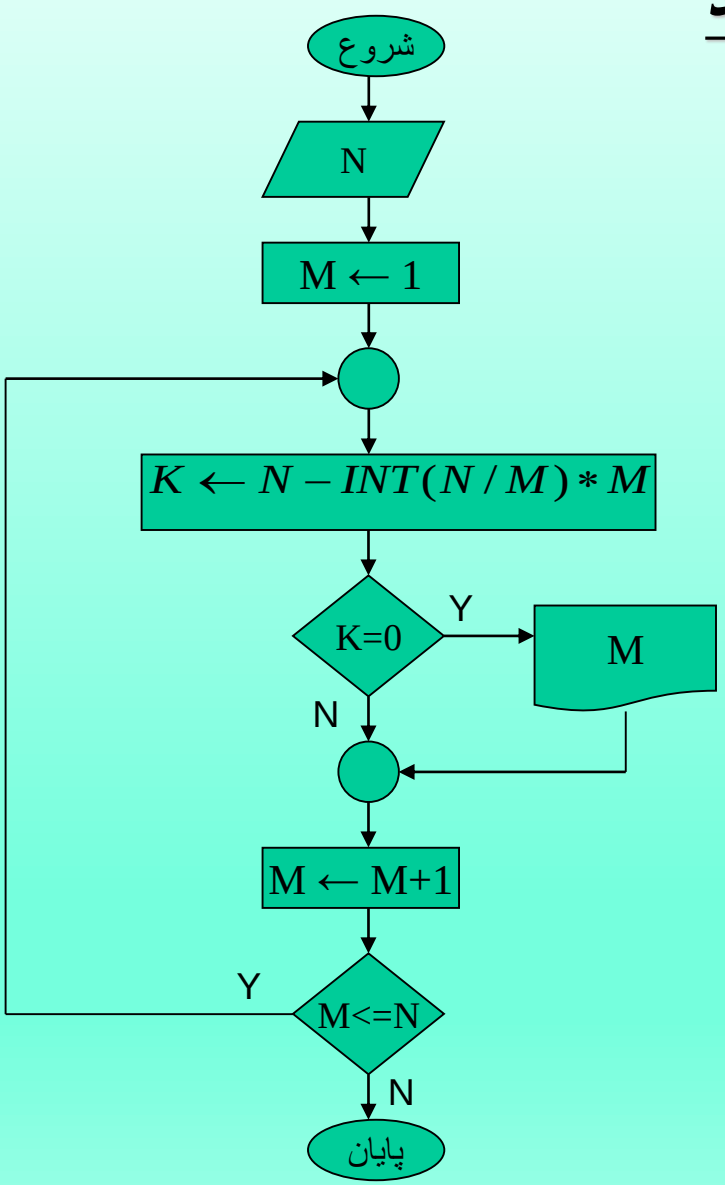
تمرین



(تعداد ارقام)

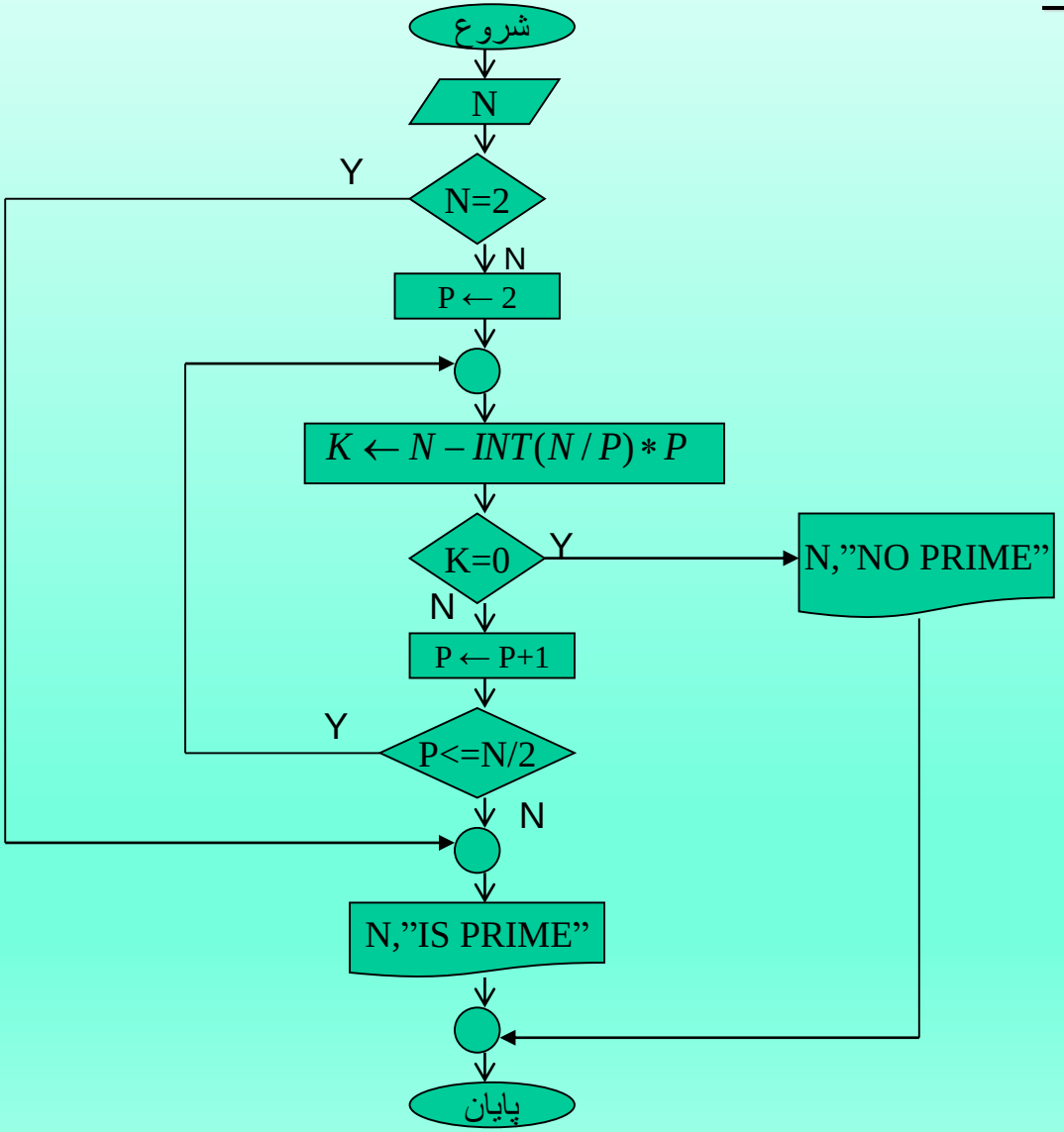
مثال ۸: الگوریتم برنامه ای را بنویسید که عدد طبیعی N>1 را خوانده و مقسوم علیه های آن را نمایش دهد

تمرین



مثال ۹: الگوریتم برنامه ای را بنویسید که یک عدد را خوانده ، اول بودن آن را تعیین نمایید

تمرین



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

تمرین

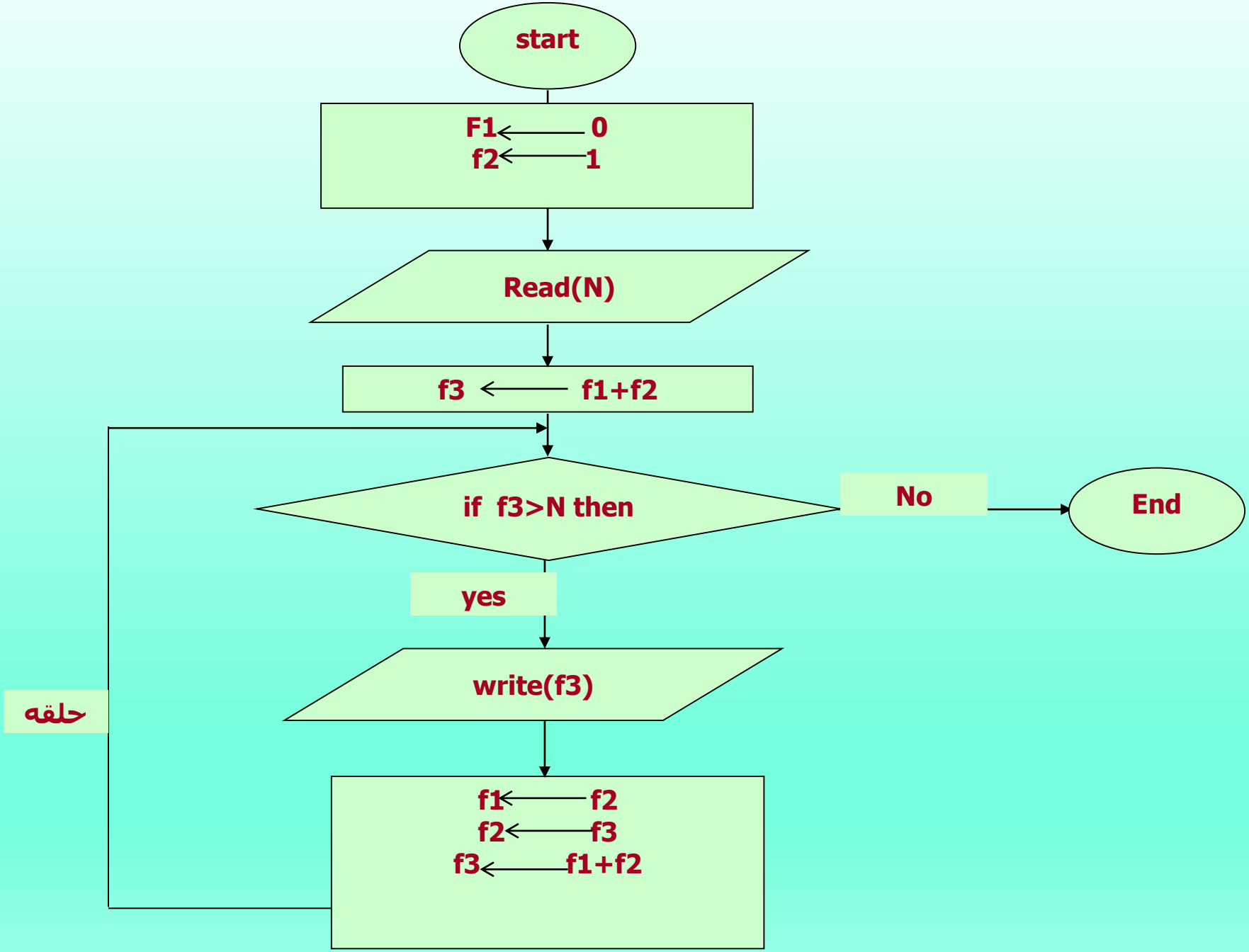
در حالت کلی جملات سری بصورت:

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

N عدد خوانده شده

F1 جمله اول سري
F2 جمله دوم سري
F3 جمله سوم سري

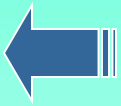
Example: 1,1,2,3,5,8,13 …

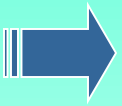

start
F1 ← 0
f2 ← 1
Read(N)
f3 ← f1+f2
if f3>N then
No
End
yes
write(f3)
حلقه
f1 ← f2
f2 ← f3
f3 ← f1+f2

برنامه نویسی به زبان سی پلاس پلاس C++

اسلاید ها

مقدمات C++





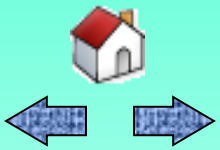
تاریخچه مختصر C++

این زبان در اوائل دهه ۱۹۸۰ توسط Bjarne stroustrup در آزمایشگاه بل طراحی شده. این زبان عملاً توسعه یافته زبان برنامه نویسی C می باشد که امکان نوشتن برنامه‌های ساخت یافته شئ گرا را می‌دهد.

قانون نامگذاری شناسه‌ها

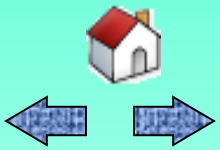
1) حروف کوچک و بزرگ در نامگذاری شناسه‌ها متفاوت می‌باشند.

بنابراین Xy ، XY ، xY ، xy چهار شناسه متفاوت از نظر C++ می‌باشد.



قانون نامگذاری شناسه‌ها

۲) در نامگذاری شناسه‌ها از حروف الفباء، ارقام وزیر خط (underscore) استفاده می‌شود و حداکثر طول شناسه ۳۱ می‌باشد و شناسه بایستی با یک رقم شروع نگردد.

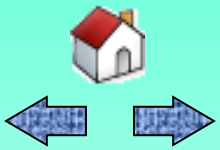


قانون نامگذاری شناسه‌ها

۳) برای نامگذاری شناسه‌ها از کلمات کلیدی نبایستی استفاده نمود. در زیر بعضی از کلمات کلیدی داده شده است.

And	Sizeof	then	xor	Template
Float	False	Friend	While	continue
extern	Private	Switch	Default	Const
delete	typedef	if	this	Virtual

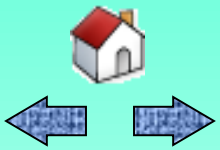
لیست کامل کلمات کلیدی



متغیرها

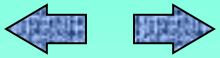
متغیر، مکانی در حافظه اصلی کامپیوتر می‌باشد که در آنجا یک مقدار را می‌توان ذخیره و در برنامه از آن استفاده نمود. قانون نامگذاری متغیرها همان قانون نامگذاری شناسه‌ها می‌باشد.

در اسلاید بعد به انواع داده ها اشاره می شود.

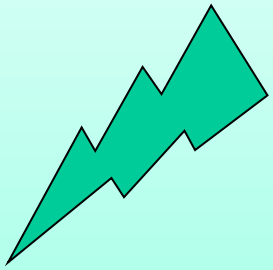


انواع داده ها

نوع داده	مقادیر	حافظه لازم
int	32767 تا 32768–	۲ بایت
unsigned int	0 تا 65535	۲بایت
long int	2147483647 تا 2147483648–	۴ بایت
unsigned long int	0 تا 4294967295	۴ بایت
char	یک کارکتر	۱بایت
unsigned char	127 تا 128–	۱ بایت
float	3.4e38 تا 1.2e-38	۴ بایت
double	1.8e308 تا 2.2e-308	۸ بایت

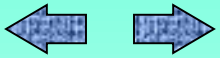


اعلان متغیرها



قبل از آنکه در برنامه به متغیرها مقـداری تخصـیص داده شود و از آنها استفاده گردد بایستی آنها را در برنامه اعلان نمود.

در اسلاید بعد مثال هایی از اعلان متغیر ذکر شده است.



چند مثال از اعلان متغیر ها :

- ✓ برای اعلان متغیرx از نوع int :

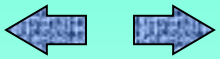
```
int    x;
```

- ✓ برای اعلان متغیرهای p و q را از نوع float که هر کدام چهار بایت از حافظه را اشغال می‌کنند :

```
float  p , q  ;
```

- ✓ برای اعلان متغیر next از نوع کرکتر که می‌توان یکی از ۲۵۶ کرکتر را به آن تخصیص داد و یک بایت را اشغال می‌کند.

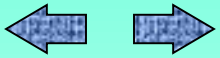
```
char next ;
```



تخصیص مقادیر به متغیرها

با استفاده از عملگر = می‌توان به متغیرها مقدار اولیه تخصیص نمود.

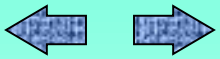
در اسلاید بعد مثال هایی از اعلان متغیر ذکر شده است.



مثال :

- در دستورالعمل `int x=26;`
  X را از نوع int با مقدار اولیه 26 اعلان نموده .

- در دستورالعمل `long a=67000 , b=260;`
  متغیرهای b و a را از نوع long int تعریف نموده با مقادیر بترتیب 260 و 67000.

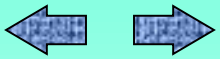


داده‌های از نوع کرکتر

برای نمایش داده‌های از نوع char در حافظه کامپیوتر از جدول ASCII استفاده می‌شود. جدول اسکی به هر یک از ۲۵۶ کرکتر یک عدد منحصر بفرد بین ۰ تا ۲۵۵ تخصیص می‌دهد.

کرکترهای مخصوص

کامپیلر C++ بعضی از کرکترهای مخصوص که در برنامه می‌توان از آنها برای فرمت بندی استفاده کرد را تشخیص می‌دهد. تعدادی از این کرکترهای مخصوص به همراه کاربرد آنها در اسلاید بعد آورده شده است .

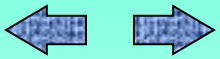


کرکترهای مخصوص

\n	Newline
\t	Tab
\b	Backspace
\a	Beep sound
\”	Double quote
\’	Single quote
\0	Null character
\?	Question mark
\\	Back slash

بعنوان مثال از کرکتر \a می‌توان برای ایجاد صدای beep استفاده نمود.

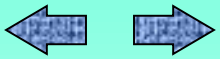
```
char        x = '\a' ;
```



رشته‌ها

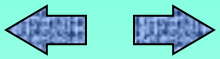
رشته یا string عبارتست از دنباله‌ای از کرکترها که بین " " قرار داده می‌شود. در حافظه کامپیوتر انتهای رشته‌ها بوسیله \0 ختم می‌گردد.

در اسلاید بعد به دو مثال دقت نمایید.



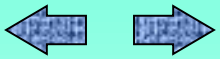
مثال ۱ :

"BOOK STORE" یک رشته ده کرکتـری می‌باشـد که با توجه به کرکتر 0\ که به انتهای آن در حافظه اضافه می شود جمعــاً یازده بایت را اشغال می‌کند.



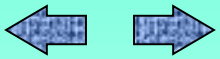
مثال ۲ :

دقت نمایید که "w" یک رشته می‌باشد که دو بایت از حافظه را اشغال می‌کند در حالیکه 'w' یک کرکتر می‌باشد که یک بایت از حافظه را اشغال می‌نماید.



نمایش مقادیر داده‌ها

برای نمایش داده‌ها بر روی صفحه مانیتور از cout که بدنبال آن عملگر درج یعنی << قید شده باشد استفاده می‌گردد. بایستی توجه داشت که دوکرکتر < پشت سر هم توسط C++ بصورت یک کرکتر تلقی می‌گردد.



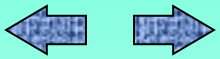
مثال :

- ✓ برای نمایش پیغام good morning بر روی صفحه نمایش :

```
cout << "good morning";
```

- ✓ برای نمایش مقدار متغیر x بر روی صفحه نمایش :

```
cout << x ;
```

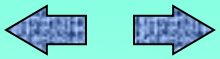


دریافت مقادیر متغیرها

به منظور دریافت مقادیر برای متغیرها در ضمن اجرای برنامه از صفحه کلید، از cin که بدنبال آن عملگر استخراج یعنی >> قید شده باشد می‌توان استفاده نمود.

مثال :

```
int x;
cout << "Enter a number:" ;
cin >> x;
```



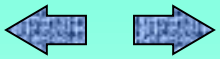
عملگر انتساب

عملگر انتساب = می‌باشد که باعث می‌گردد مقدار عبارت در طرف راست این عملگر ارزیابی شده و در متغیر طرف چپ آن قرار گیرد.

مثال :

```
x=a+b;
x=35 ;
x=y=z=26 ;
```

از عملگرهای انتساب چندگانه نیز می‌توان استفاده نمود. که مقدار سه متغیر z و y و x برابر با 26 میشود.

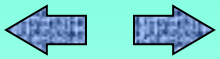


عملگرهای محاسباتی

در C++ پنج عملگر محاسباتی وجود دارد که عبارتند از :

+	جمع
–	تفریق
*	ضرب
/	تقسیم
٪	باقیمانده

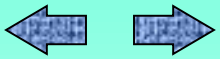
این عملگرها دو تائی می‌باشند زیرا روی دو عملوند عمل می‌نمایند. از طرف دیگر عملگرهای + و – رامی‌توان بعنوان عملگرهای یکتائی نیز در نظر گرفت.



مثال ۱ :

در حالتی که هر دو عملوند عملگرهای ٪ ، / ، * ، + ، – از نوع صحیح باشد نتیجه عمل از نوع صحیح می‌باشد.

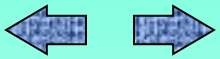
عبارت	نتیجه
5 + 2	7
5 * 2	10
5 – 2	3
5 % 2	1
5 / 2	2



مثال ۲ :

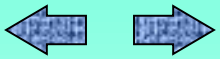
در صورتیکه حداقل یکی از عملوندهای عملگرهای / ، * ، – ، + از نوع اعشاری باشد نتیجه عمل از نوع اعشاری می‌باشد.

عبارت	نتیجه
5. 0 +2	7. 0
5* 2. 0	10.0
5. 0 / 2	2.5
5. 0 -2	3. 0
5. 0 / 2. 0	2.5



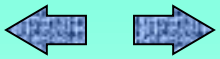
عملگرهای افزایش و کاهش

در C++ ، افزایش یک واحد به مقدار یک متغیر از نوع صحیح را افزایش و بطور مشابه کاهش یک واحد از مقدار یک متغیر از نوع صحیح را کاهش می‌نامند..



عملگرهای افزایش و کاهش

عملگر کاهش را با – – و عملگر افزایش را بـــا ++ نمـــایش می‌دهنـــد. چـــون عملگرهـــای ++ و – – فقـــط روی یـــک عملوند اثر دارند این دو عملگر نیز جـــزء عملگرهای یکتائی می‌باشند.



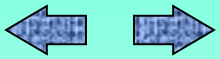
مثال :

سه دستور العمل :

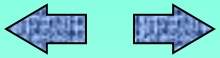
```
++x;
x++;
x=x+1;
```

معادل می‌باشند و بطریق مشابه سه دستورالعمل زیر نیز معادل می باشند.

```
- - y ;
y=y-1;
y- - ;
```



از عملگرهای ++ و -- می‌توان بدو صورت پیشوندی و پسوندی استفاده نمود.
در دستورالعمل‌های پیچیده عملگر پیشوندی قبل از انتساب ارزیابی میشود و عملگر پسوندی بعد از انتساب ارزیابی می‌شود.

مثال :

```
int x=5;
y=++x * 2;
```

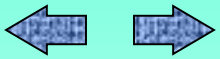
پس از اجرای دستورالعملهای فوق :

y=12

```
int  x=5;
y=x++ * 2;
```

پس از اجرای دستورالعملهای فوق :

y=12



عملگر sizeof

Sizeof ازعملگرهای یکتائی می باشد و مشخص کننده تعداد بایت هائی است که یک نوع داده اشغال می‌کند.

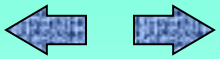
مثال :

```
int   x;
cout << sizeof x ;
```

مقدار ۲ نمایش داده می‌شود .

```
cout << sizeof(float) ;
```

مقدار ۴ نمایش داده می شود.



عملگرهای جایگزینی محاسباتی

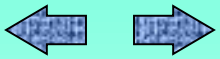
برای ساده‌تر نوشتن عبارتها در C++ ، می‌توان از عملگرهای جایگزینی محاسبـــــاتی استفاده نمود.

+= -= *= /= %=

اولویت عملگرها

ارزیابی مقدار یک عبارت ریاضی براساس جدول اولویت عملگرها انجام می‌گردد. در ذیل جدول اولویت عملگرها براساس بترتیب از بیشترین اولویت به کمترین اولویت داده شده است.

چپ به راست	پرانتزها	( )
راست به چپ	عملگرهای یکتایی	sizeof ++ -- + -
چپ به راست	عملگرهای ضرب و تقسیم و باقیمانده	* / %
چپ به راست	عملگرهای جمع و تفریق	+ –
چپ به راست	عملگرهای درج و استخراج	>> <<
راست به چپ	عملگرهای جایگزینی و انتساب	= += -= *= /= %=



مثال ۱ :

(5+2) *(6+2*2)/2

با توجه به جدول اولویت عملگرها داریم که

7 *(6+2*2)/2
7*(6+4)/2
7* 10 /2
70 /2
35

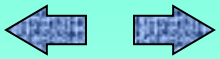
مثال ۲ :

```
int a=6 , b=2, c=8, d=12;
d=a++ * b/c ++;
cout << d << c << b << a;
```

خروجی :

توضیحات (Comments)

توضیحات در برنامه باعث خوانائی بیشتر و درک بهتر برنامه میشود. بنابراین توصیه بر آن است که حتی الامکان در برنامه‌ها از توضیحات استفاده نمائیم. در C++، توضیحات بدو صورت انجام می‌گیرد که در اسلایدهای بعد به آن اشاره شده است.



توضیحات (Comments)

الف: این نوع توضیح بوسیله // انجام می‌شود. که کامپیوتر هر چیــزی را که بعد از // قرار داده شود تا انتهای آن خط اغماض می‌نماید.

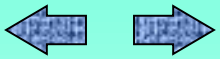
مثال :

```
c=a+b;//c is equal to sum of a  and b
```

ب:توضیح نوع دوم با /* شروع شده و به */ ختم می‌شود و هر چیزی که بین* / و */ قرار گیرد اغماض می‌نماید .

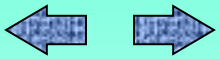
مثال :

```
/ * this is a program
to calcufate sum of
n integer numbers */
```



توابع کتابخانه

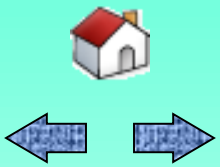
زبان C++ مجهز به تعدادی توابع کتابخانه می‌باشد. بعنوان مثال تعدادی توابع کتابخانه برای عملیات ورودی و خروجی وجود دارند. معمولاً توابع کتابخانه مشابه ، بصورت برنامه‌های هدف (برنامه ترجمه شده بزبان ماشین) در قالب فایلهای کتابخانه دسته بندی و مورد استفاده قرا رمی‌گیرند. این فایلها را فایلهای header می‌نامند و دارای پسوند h. می‌باشند.



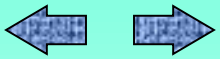
نحوه استفاده از توابع کتابخانه ای

برای استفاده از توابع کتابخانه خاصی بایستـــــی نام فایل header آنرا در ابتدای برنامه در دستور #include قرار دهیم.

#include < اسم فایل header >



تابع	نوع	شرح	فایل هیدر
abs(i)	*int*	قدرمطلق i	stdlib.h
cos(d)	*double*	کسینوس d	math.h
exp(d)	*double*	e^x	math.h
log(d)	*double*	$\log_e$ d	math.h
log10(d)	*double*	Log_{10} d	math.h
sin(d)	*double*	سینوس d	math.h
sqrt(d)	*double*	جذر d	math.h
strlen(s)	*int*	تعداد کرکترهای رشته s	string.h
tan(d)	*double*	تانژانت d	math.h
toascii(c)	*int*	کداسکی کرکتر c	stdlib.h
tolower(c)	*int*	تبدیل به حروف کوچک	stdlib.h
toupper(c)	*int*	تبدیل به حرف بزرگ	stdlib.h
Cout-Cin		ورودی – خروجی	iostream.h

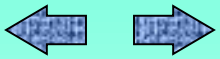


برنامه در C++

اکنون باتوجه به مطالب گفته شده قادر خواهیم بود که تعدادی برنامه ساده و کوچک به زبان C++ بنویسیم. برای نوشتن برنامه بایستی دستورالعملها را در تابع ( )main قرار دهیم و برای اینکار می‌توان به یکی از دو طریقی که در اسلایدهای بعد آمده است ، عمل نمود.

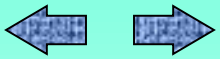
روش اول :

```
#include      <       >
int main( )
{
1 دستورالعمل ;
2 دستورالعمل ;
.
.
.
n  دستورالعمل;
return 0 ;
}
```

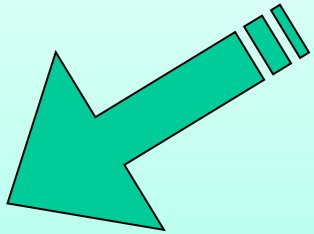


روش دوم :

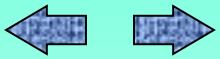
```
#include     <      >
void main( )
{
1 دستورالعمل ;
2 دستورالعمل ;
.
.
.
n  دستورالعمل;
}
```



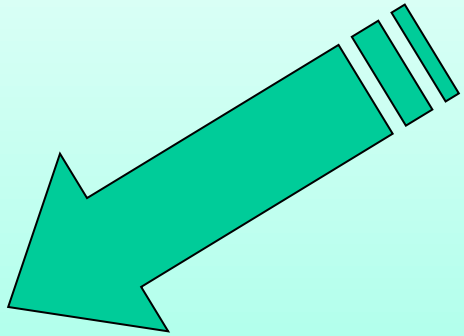
برنامه ای که پیغام C++ is an object oriented language را روی صفحه مانیتور نمایش می دهد.



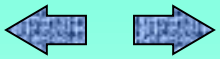
```
#include    <iostream.h>
int main( )
{
cout <<"C++ is an object oriented language \n" ;
return 0 ;
}
```



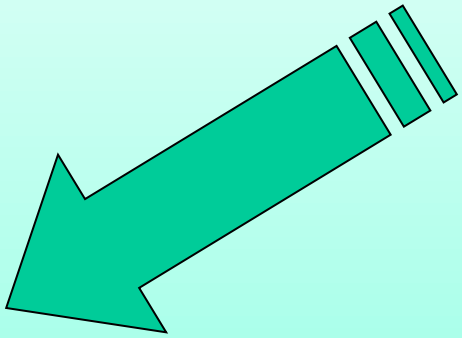
برنامه زیر یک حرف انگلیسی کوچک را گرفته به حرف بزرگ تبدیل می‌نماید.



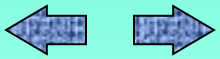
```
#include    <iostream.h>
#include    <stdlib. h>
int main( )
{
 char c1 , c2;
cout << "Enter a lowercase letter:"
cin >> c1;
c2 = toupper(c1);
cout << c2 << endl;
return 0; }
```



دو عدد از نوع اعشاری را گرفته مجموع و حاصلضرب آنها را محاسبه و نمایش می‌دهد.



```
#include    <iostream.h>
int main( )
{
float         x,y,s,p ;
cin >> x >> y ;
s= x+y ;
p=x*y;
cout << s <<endl << p;
return 0 ;
}
```



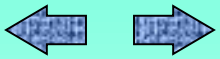
ساختارهای تصمیم گیری و تکرار

1. عملگر های رابطه ای
2. عملگر شرطی
3. دستورالعمل شرطی
4. عملگر کاما
5. عملگر های منطقی
6. دستورالعمل For

عملگرهای رابطه ای

از این عملگرها برای تعیین اینکه آیا دو عدد با هم معادلند یا یکی از دیگری بزرگتر یا کوچکتر می‌باشد استفاده می‌گردد. عملگرهای رابطه‌ای عبارتند از:

مساوي	= =
مخالف	! =
بزرگتر	>
بزرگتر یا مساوي	> =
کوچکتر	<
کوچکتر یا مساوي	<=

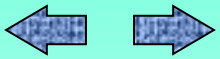


عملگر شرطی

شکل کلی عملگر شرطی بصورت زیر می‌باشد:

```
expression _ test ? expression _ true : expression _ false
```

عملگر شرطی تنها عملگری در C++ می‌باشد که دارای سه عملوند می‌باشد.



مثال ۱ :

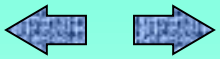
```
int x=10,y=20,b;
b=(x>y) ? x : y ;
```

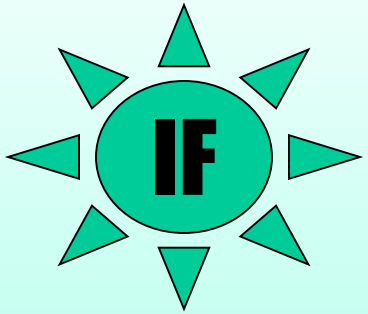
این دو دستور العمل باعث میشوند که ماکزیمم مقادیر y و x در b قرار بگیرد.

مثال ۲ :

```
x>=10 ? cout << "passed" : cout << "failed" ;
```

اگر مقدار x بزرگتر یا مساوی ده باشد رشتهٔ passed در غیر اینصورت رشته failed نمایش داده میشود.

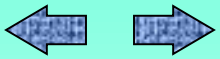




دستورالعمل شرطی

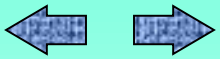
توسط این دستور شرطی را تست نموده و بسته به آنکه شرط درست یـا غلـط باشـد عکس‌العمـل خاصی را نشان دهیم.

```
if(   عبارت )
{
        دستورالعمل 1 ;
              .
        دستورالعمل n;
 }
 else
 {
        دستورالعمل 1 ;
              .
        دستورالعمل n;
 }
```



مثال ١ :

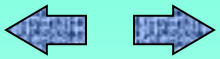
```
if(x != y)
{
cout << x ;
++ x ;
}
else
        {
cout << y ;
- - y ;
         }
```



مثال ۲:

برنامه زیر یک عدد اعشاری را از ورودی گرفته جذر آنرا محاسبه می‌نماید.

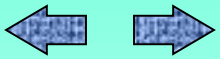
```
#include    <iostream.h>
#include    <math . h>
int main( )
{
float  x,s;
cin >> x ;
if( x < 0 )
cout << " x is negative" << endl ;
else
{
s = sqrt(x) ;
cout << s << endl ;
                    }
return 0;
}
```



عملگر کاما

تعدادی عبارت را می‌توان با کاما بهم متصل نمود و تشکیل یک عبارت پیچیده‌تری را داد. این عبارتها به ترتیب از چپ به راست ارزیابی شده و مقدار عبارت معادل عبارت n می‌باشد.

(عبارت n , ... , عبارت 3 , عبارت 2 , عبارت 1)

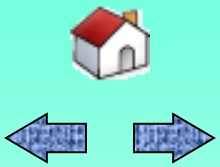


مثال :

اگر داشته باشیم int a=2 , b=4 , c=5 ; عبارت زیر را در نظر بگیرید:

(++ a , a+b, ++ c, c+b)

مقدار عبارت برابر است با b+c که معادل 10 می‌باشد.



عملگرهای منطقی

با استفاده از عملگرهای منطقی می‌توان شرطهای ترکیبی در برنامه ایجاد نمود.
عملگرهای منطقی عبارتست از :

AND

OR

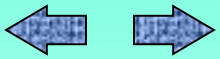
NOT

که در C++ به ترتیب بصورت زیر نشان داده میشود.

&&

||

!



جدول درستی سه عملگر شرطی

And

a	b	a && b
true	true	True
true	false	False
false	true	False
false	false	False

Or

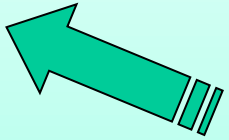
a	b	a || b
true	true	True
true	false	True
false	true	True
false	false	False

Not

a	!a
true	False
false	True

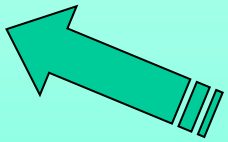
چند مثال :

```
if ((x= = 5) ||(y != 0))
    cout << x << endl ;
```

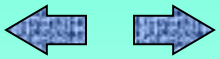


اگر x برابر با 5 یا y مخالف صفر باشد مقدار x نمایش داده شود .

```
if(x)
 x = 0 ;
```

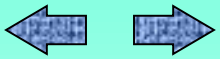


اگر مقدار x مخالف صفر باشد، آنگاه x برابر با صفر شود .



برنامه زیر طول سه پاره‌خط را از ورودی گرفته مشخص می‌نماید که آیا تشکیل یک مثلث میدهد یا خیر؟

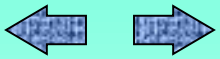
```
#include       < iostream.h >
int main( )
{
float  a, b, c;
cout << "Enter three real numbers" << endl ;
cin >> a >> b >> c; //
if(( a < b + c) &&(b < a+c) &&(c < a+b))
cout << "It is a triangle" ;
else
cout << "Not a triangle" ;
return  0 ;
}
```



دستورالعمل For

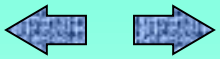
از دستور العمل for برای تکرار دستورالعملها استفاده میشود. شکل کلی دستور for بصورت زیر می‌باشد:

```
for (عبارت 1 ; عبارت 2 ; عبارت 3)
 {
  دستورالعمل 1 ;
  دستورالعمل 2 ;
  .
  .
  .
  دستورالعمل n;
 }
```



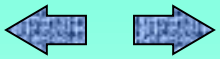
برنامه زیر عدد صحیح و مثبت n را از ورودی گرفته فاکتوریل آنرا محاسبه ونمایش می‌دهد.

```
#include          <iostream.h>
int      main( )
{
int  n, i ;
long fact = 1 ;
cout << "Enter a positive integer number";
cin >> n;
for( i=1; i<=n; ++i) fact *= i;
 cout << fact << endl;
return 0 ;
}
```



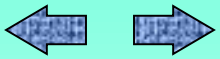
برنامه زیر مجموع اعداد صحیح و متوالی بین ۱ تا n را محاسبه نموده و نمایش می‌دهد.

```
#include        <iostream.h>
int     main( )
{
int n, i=1 ;
long s = 0 ;
cin >> n ;
for(; i<=n; i++) s += i;
cout << s ;
return  0 ; }
```



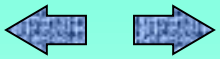
برنامه زیر ارقام 0 تا 9 را نمایش می‌دهد.

```
#include    <iostream.h>
int main( )
{
int  j=0 ;
for( ; j <= 9 ; ) cout << j++ << endl;
return 0 ;
}
```



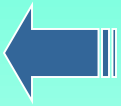
برنامه زیر کلیه اعداد سه رقمی که با ارقام 1 ، 2 ، 3 ایجاد می‌شوند را نمایش می‌دهد.

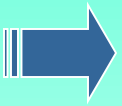
```
#include    <iostream.h>
int main( )
{
int  i,j,k,n;
for(i=1; i<=3; ++i)
for(j=1; j<=3; ++j)
for(k=1; k<=3; ++k)
{
n=i*100 + j*10+k;
cout << n << '\n' ;
}
return 0 ;
}
```



فصل سوم

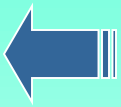
سایر ساختارهای تکرار





حلقه ها

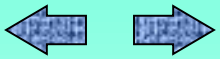
1. دستورالعمل while
2. دستورالعمل do while
3. دستورالعمل break
4. دستورالعمل continue
5. دستورالعمل switch
6. تابع cin.get()
7. عملگر static_cast<>()
8. جدول اولویت عملگرها



دستورالعمل while

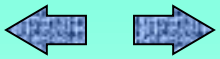
از این دستور العمل مانند دستورالعمل for برای تکرار یک دستورالعمل ساده یا ترکیبی استفاده می‌گردد. شکل کلی این دستور العمل بصورت زیر می‌باشد.

```
while( شرط)
{
 دستورالعمل ۱ ;
 دستورالعمل ۲ ;
    .
    .
 دستورالعمل n ;
}
```



تفاوت دستورهای for وwhile

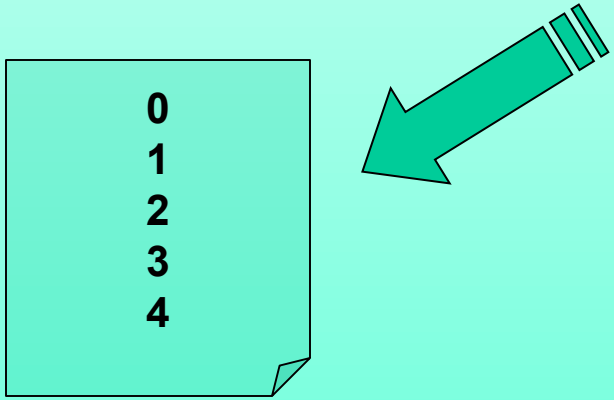
دستورالعمل for زمانی استفاده میشود که تعداد دفعات تکرار از قبل مشخص و معین باشد. در صورتیکه تعداد دفعات تکرار مشخص نباشد بایستی از دستورالعمل while استفاده نمود.



مثال :

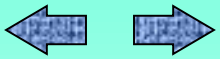
```
int x=0
while(x<5)
cout << x ++<< endl;
```

با اجرای قطعه برنامه فوق مقادیر زیر نمایش داده میشود :



برنامهٔ فوق n مقدار از نوع اعشاری را گرفته میانگین آنها را محاسبه و در متغیر avg قرار می‌دهد.

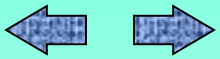
```
#inclode   <iostream.h>
int main( )
{
int count = 0 , n;
float  x, sum = 0 , avg ;
cin >> n ; /*   تعداد مقادیر ورودی n*/
while(count < n){
cin >> x ;
sum += x ;
++ count ; }
avg = sum / n ;
cout << avg << endl;
return 0 ; }
```



دستورالعمل do while

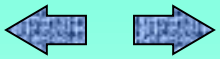
این دستور العمل نیز برای تکرار یک دستورالعمل ساده یا ترکیبی استفاده می‌شود. شکل کلی این دستورالعمل بصورت زیر می‌باشد.

```
do
{
 دستورالعمل ۱ ;
 دستورالعمل ۲ ;
    .
    .
 دستورالعمل n ;
} while( شرط);
```



تفاوت دستورهای do while و while

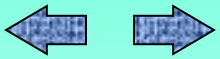
در دستورالعمل **while** ابتدا مقدار شرط ارزیابی شده اما در دستورالعمل **do while** ابتدا دستورالعمل اجرا شده سپس مقدار شرط ارزیابی می‌گردد. بنابراین دستورالعمل **do while** حداقل یک بار انجام میشود .



مثال :

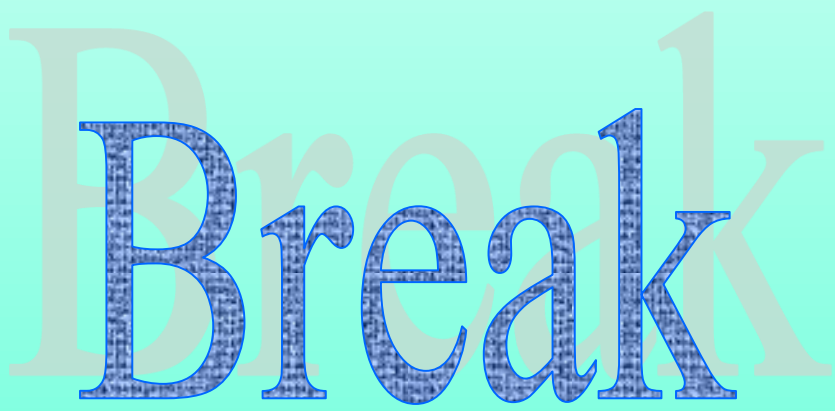
```
#include   <iostream.h>
int main( )
{
int count = 0;
do
cout << count ++<<endl ;
while(count <= 9);
return 0 ; }
```

ارقام 0 تا 9 را روی ده خط نمایش می‌دهد



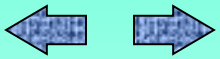
دستورالعمل break

این دستورالعمل باعث توقف دستورالعملهای تکرار(for , while ,do while) شده و کنترل به خارج از این دستورالعملها منتقل می‌نماید.



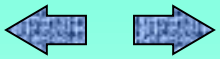
مثال ١ :

```
#include   <iostream.h>
int main( )
{
float x, s=0.0 ;
cin >> x ;
while(x <= 1000.0) {
if(x < 0.0){
cout << "Error-Negative Value" ;
break;
}
s += x ;
cin >> x ;}
cout << s << endl ;
return 0 ; }
```



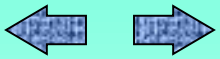
مثال ٢:

```
#include   <iostream.h>
int main( )
{
int count = 0 ;
while( 1 )
{
count ++ ;
if(count > 10 )
break ;
}
cout << "counter :    " << count << "\n";
return 0 ;
}
```



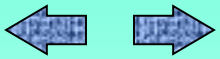
مثال ٣:

```
#include   <iostream.h>
void  main( )
{
int count;
float x, sum = 0;
cin >> x ;
for(count = 1; x < 1000 . 0; ++ count )
{
cin >> x ;
if(x < 0.0) {
cout << "Error – Negative value " <<endl;
break ;
}
sum += x ; }
cout << sum << '\n' ; }
```



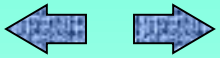
مثال ۴:

```
#include   <iostream.h>
int main( )
{
float x , sum = 0.0 ;
do
{
cin >> x ;
if(x < 0.0)
     {
cout << "Error – Negative Value" << endl ;
break ;
     }
sum += x ;
} while(x <= 1000.0);
cout << sum << endl ;
return 0 ; }
```



جدول اولویت عملگرها

چپ به راست	( )
راست به چپ	Static_cast < >( ) ++ -- + - sizeof
چپ به راست	* / %
چپ به راست	+ -
چپ به راست	<< >>
چپ به راست	< <= > >=
چپ به راست	= = !=
راست به چپ	? :
راست به چپ	= += -= *= /= %=
چپ به راست	,



تمرینات سری اول

1. برنامه‌ای بنویسید که دوعدد "50+100" را باهم جمع کند و در خروجی نمایش دهد؟
2. برنامه‌ای بنویسید که دو صحیح عدد از ورودی دریافت کند و حاصل جمع، ضرب، تفاضل و تقیسم آنها را محاسبه کند و بصورت جداگانه همه آنها نمایش دهد؟
3. برنامه‌ای بنویسید که طول و عرض مستطیل را از ورودی گرفته و محیط و مساحت آن را محاسبه و نمایش دهد؟
4. برنامه‌ای بنویسید که شعاع یک دایره را از ورودی دریافت کرده محیط و مساحت آن را محاسبه و نمایش دهد؟(mohit=2*(p*r) , masahat=p*r^2)
5. برنامه‌ای بنویسید که حجم یک مکعب مستطیل را محاسبه کند؟ (R=a*b*h)
6. برنامه‌ای بنویسید که دو عدد را از ورودی خوانده سپس مقادیر آن دو را با هم جابجا کند؟

تمرینات سری دوم

عبارات شرطی

1. برنامه‌ای بنویسید که یک عدد از ورودي بگیرد و زوج یا فرد بودن آن را مشخص کند؟
2. برنامه‌ای بنویسید که دو عدد از ورودي دریافت کرده بزرگترین عدد را پیدا کرده در خروجي نمایش نماید.
3. برنامه‌ای بنویسید که سه عدد از ورودي دریافت کرده کوچکترین عدد را پیدا کرده در خروجي نمایش نماید.

تمرینات سری سوم

حلقه ها

1. برنامه ای بنویسید که 100 عدد از ورودی خوانده مجموع آنها را محاسبه و چاپ دهد.
2. برنامه ای بنویسید که که یک عدد بزرگتر از صفر را خوانده سپس به تعداد آن عدد، اعداد دیگری را خوانده مجموع و میانگین آنها را نمایش دهد.
3. برنامه ای بنویسید که 10 عدد را گرفته و تعیین کند کدام زوج و کدام فرد است.
4. برنامه‌ای بنویسید که یک عدد از ورودي بگيرد و فاکتوريل آن را محاسبه کرده و خروجي نمايش مي دهد؟
5. برنامه ای بنویسید که یک عدد مثبت را خوانده و تعداد ارقام عدد را نشان دهد.
6. برنامه ای بنویسید که یک عدد را از ورودی خوانده ، اول بودن آن را تعیین نمایید
7. برنامه ای بنویسید که عددي از ورودي دريافت کرده، سري فيبوناچي قبل از آنرا تا آن عدد توليد نمايد.

پایان