

بخش اول: مبانی و مفاهیم کامپیوتر

۱،۱-مقدمه

امروزه کامپیوتر به عنوان ابزار قدرتمندی در زمینه های مختلف مورد استفاده قرار می گیرد و به جرات می توان بیان کرد که انجام بسیاری از فعالیت های پژوهشی بدون حضور کامپیوتر سخت و گاهی غیر ممکن می باشد. استفاده از کامپیوتر بخصوص در چند دهه ی اخیر منجر به فعالیت های بسیاری شده است که از جمله آنها می توان به اینترنت ، شبکه های محاسباتی گرید و پروژه ژنوم انسان و... اشاره کرد. اینترنت امکان دسترسی از راه دور به کامپیوتر دیگر و انباره های اطلاعات در هر جای دنیا که باشند را به کاربران کامپیوتر می دهد. شبکه های محاسباتی گرید مجموعه ای از چندین سیستم با قدرت محاسباتی متفاوت می باشد که با متصل شدن این قدرت محاسباتی حاصل یک ابر رایانه ی مجازی شکل میگیرد که با استفاده از آن می توان بسیاری از محاسبات پیچیده ی ریاضی، نجوم، زیست و... در زمان بسیار کمی انجام داد. شعار شبکه های محاسباتی نادیده گرفتن نگرانی های ناشی از محدودیت سخت افزاری سیستم های کامپیوتری می باشد. در پروژه ژنوم انسان با استفاده از برنامه های کامپیوتری توانستند توالی ژن های انسان را بدست آورند و آنها را در پایگاه داده ای نگهداری کنند و برای تحقیقات بیشتری از آن استفاده کنند. کامپیوترها از زمان پیدایش خود تا کنون نسل های مختلفی را سپری کرده اند که خالی از لطف نیست که نگاه کوتاهی بر آنها داشته باشیم.

کامپیوترهای نسل اول : این کامپیوترها که در اوایل دهه ۱۹۵۰ ساخته شدند از لامپ خلاء بعنوان جزو اصلی خود استفاده می کردند که در نتیجه حجم بسیار بالایی داشته و انرژی بالایی را نیز مصرف می نمودند. انیاک (Eniac) یکی از کامپیوترهای معروف این دوره بود.

کامپیوترهای نسل دوم : که در اوایل دهه ۱۹۶۰ ابداع گردیدند. ویژگی مهم آنها استفاده از ترانزیستور بجای لامپ خلاء بود که باعث کاهش اندازه کامپیوترها گردید.

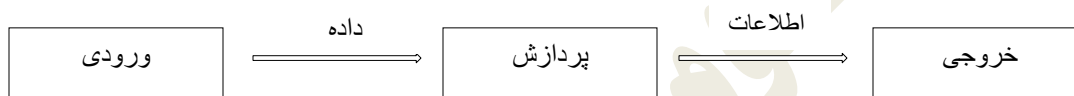
کامپیوترهای نسل سوم : این کامپیوترها در سال ۱۹۶۴ با ابداع مدارات مجتمع IC که صدها ترانزیستور را در یک فضای کوچک جای می داد، ایجاد شدند. ابداع مدارات مجتمع باعث بالا رفتن سرعت و کاهش بیشتر حجم کامپیوترها گردید.

نسل چهارم کامپیوترها : در اواسط دهه ۱۹۷۰ با ابداع مدارات مجتمع با فشردگی بالا، حجم کامپیوترها بازم کاهش یافت و پای آنها را به کاربردهای خانگی و اداری باز کرد.

نسل پنجم کامپیوترها: یا نسل کامپیوترهای هوشمند که قادر به انجام اعمالی همانند استنتاج و استدلال مانند انسانها باشند. این نسل هنوز تا رسیدن به وضعیت ایده آل راه درازی دارد.

سیستم کامپیوتری :

هر سیستم کامپیوتری از مجموعه ای از سخت افزارها و نرم افزارهایی تشکیل شده است که در جهت انجام کار خاصی با یکدیگر همکاری می کنند. تعریف دقیق تر آن بدین صورت است که هر سیستم کامپیوتری از وسایل الکترونیکی و الکترومکانیکی تشکیل شده است که داده هایی را به عنوان ورودی دریافت کرده و عملیات خاصی را طبق مجموعه ای از دستورالعمل ها بر روی داده ها انجام می دهد و نتایج حاصل از عملیات را به عنوان خروجی تولید می کند. شکل ۱ نمایی از یک سیستم کامپیوتری را نشان می دهد.



شکل ۱: نمایی از یک سیستم کامپیوتری

داده^۱: مجموعه مطالبی که وارد کامپیوتر می شود داده گفته می شود. داده ها می توانند به صورت عدد، حرف، صدا، تصویر و ... باشد.

پردازش^۲: به کلیه فعالیت های صورت گرفته بر روی داده ها که منجر به پیدایش نتایج می شود پردازش داده ها گفته می شود.

اطلاعات^۳: به خروجی که بعد از پردازش داده ها تولید می شود اطلاعات اطلاق می گردد.

الگوریتم^۴: دستورالعملهایی که برای کامپیوتر نوشته می شود را الگوریتم گوئیم .

برنامه کامپیوتری^۵: به تشریح الگوریتم ها برای کامپیوتر با استفاده از یک زبان برنامه سازی گفته می شود.

زبان برنامه سازی^۶:

¹ Data
² Processing
³ Information
⁴ Algorithm
⁵ Computer Prodraming
⁶ Prodraming Languages

زبانی است که برای کامپیوتر قابل فهم بوده و الگوریتمها با استفاده از آن به کامپیوتر داده می شوند. مهندسين نرم افزار تاکنون برای زبانها پنج نسل را در نظر گرفته اند:

زبان نسل اول⁷: که به آن زبان ماشین نیز گفته می شود، مستقیماً به زبان خود کامپیوتر (یعنی زبان صفر و یک)

نوشته می شود و توسط کامپیوتر قابل اجرا می باشد. هر کامپیوتری زبان ماشین⁸ مخصوص به خود را دارد که وابسته به سخت افزار خود آن کامپیوتر است. به عنوان مثال قطعه کد زیر می تواند اینگونه تعبیر شود که `basepay` و `overpay` را باهم جمع کن و حاصل آن را در `grosspay` ذخیره کن:

```
+1300042774
```

```
+1400593419
```

```
+1200274027
```

زبان نسل دوم⁹: زبان اسمبلی¹⁰ است که حالت نمادین زبان ماشین است و در آن دستورات با استفاده از یک نماد بجای

بجای کد صفر و یک نوشته می شوند. کد اسمبلی مثال بالا بصورت زیر است:

```
load basepay
```

```
add overpay
```

```
store grosspay
```

نکته: زبان ماشین و زبان اسمبلی جزء زبانهای سطح پایین هستند.

زبان نسل سوم¹¹: این نسل شامل زبان های سطح بالا است که از جمله زبان های این نسل می توان به زبان های C،

C++، C#، PASCAL، Basic، FORTRAN، JAVA و... اشاره کرد. برنامه نویسی به این زبانها بسیار نزدیک به زبان انسان

هستند و از دستوراتی مشابه زبان طبیعی (اغلباً زبان انگلیسی) تشکیل شده اند. برای مثال بالا داریم:

```
grosspay = basepay + overpay
```

⁷ first-generation language (1GL)

⁸ machine language

⁹ second-generation language (2GL)

¹⁰ assembly language

¹¹ Third-generation languages (3GLs)

زبان نسل چهارم^{۱۲}: این نسل شامل زبانهای بسیار سطح بالا هست که از جمله زبان های این نسل می توان به SQL

^{۱۳} اشاره کرد که زبانی است خاص منظوره و همچون زبان طبیعی که برای دریافت اطلاعات از پایگاه داده^{۱۴} بکار می رود. در زیر

نمونه ای از دستورات این زبان ذکر شده است:

```
SELECT *
```

```
FROM Customers
```

```
WHERE Balance > 50
```

با اجرای این دستورات کلیه اطلاعات مشتریان پایگاه داده که موجودی حساب آنها بیشتر از ۵۰ دلار است نمایش داده می شود.

زبان نسل پنجم^{۱۵}: زبان های این نسل شامل محیط های گرافیکی مناسب و راحتی برای تولید نرم افزار ها هستند از

جمله می توان به Visual C++، Visual C# و ... اشاره کرد.

۱.۳-انواع کامپیوترها

کامپیوتر ها از لحاظ قدرت پردازشی به گروه های مختلفی تقسیم می شوند:

ابر رایانه ها^{۱۶}: این نوع رایانه ها، قوی ترین و گرانترین نوع رایانه ها است که از قوت اجرایی و سرعت بسیار بالایی

برخوردار هستند و بیشتر در زمینه های نظامی، تحقیقاتی، علوم فضایی و پروژه های علمی بزرگ مورد استفاده قرار می گیرند.

کامپیوترهای بزرگ^{۱۷}: این کامپیوترها از سرعت و قدرت بالایی برخوردارند و معمولا در دانشگاهها و سازمانهای بزرگ و

برای محاسبات سنگین استفاده می شوند. توان محاسباتی این رده نسبت به ابر رایانه ها کمتر است.

کامپیوترهای کوچک^{۱۸}: از آنجا که قیمت کامپیوترهای بزرگ بسیار بالا بود، در اواخر دهه ۱۹۵۰ کامپیوترهای کوچک

وارد بازار شدند که توان محاسباتی کمتری داشتند و توسط سازمانهای کوچکتر مورد استفاده قرار می گرفتند.

¹² Fourth-generation languages (4GLs)

¹³ Sequential Query Languages

¹⁴ Database

¹⁵ Fifth-generation languages (5GLs)

¹⁶ Super Computers

¹⁷ mainframe

¹⁸ minicomputer

ریز کامپیوتر^{۱۹} : در آغاز دهه ۱۹۸۰ ریز کامپیوترها یا کامپیوترهای شخصی با قیمت پایین و حجم بسیار کوچک وارد بازار شدند و مورد استقبال مردم و افراد عادی قرار گرفتند.

۱-۴- اجزای کامپیوتر:

هر کامپیوتر از دو قسمت اصلی تشکیل شده است :

- **سخت افزار^{۲۰}** : کلیه دستگاههای الکتریکی، الکترونیکی و مکانیکی تشکیل دهنده یک کامپیوتر را سخت افزار آن می گوئیم.
- **نرم افزار^{۲۱}** : مجموعه برنامه هایی هستند که برای یک کاربرد خاص نوشته شده اند و بدون آنها سخت افزار قادر به کاری نیست.

که هر یک را بطور دقیقتر مورد بررسی قرار می دهیم.

سخت افزار

اجزای تشکیل دهنده سخت افزار کامپیوتر عبارتند از : واحد ورودی، واحد خروجی، واحد حافظه، واحد محاسبه و منطق، واحد کنترل و حافظه جانبی. نحوی ارتباط این واحد ها در شکل 2 نمایش داده شده است.

واحد ورودی^{۲۲} : وظیفه این بخش دریافت داده ها از دستگاه های ورودی و انتقال آنها و تبدیل آنها به داده های قابل فهم برای کامپیوتر می باشد. دستگاههای ورودی مهم عبارتند از : صفحه کلید^{۲۳}، ماوس^{۲۴}، صفحه لمسی^{۲۵}، قلم نوری، اسکرین^{۲۶} و ...

واحد خروجی^{۲۷} : این بخش وظیفه انتقال اطلاعات از کامپیوتر به محیط خارج را بعهده دارد و مهمترین دستگاههای

خروجی عبارتند از : صفحه نمایش^{۲۸}، چاپگر^{۲۹}، رسام^{۳۰}، بلندگو^{۳۱} و ...

¹⁹ microcomputer

²⁰ Hardware

²¹ Software

²² Input Unit

²³ Keyboard

²⁴ Mouse

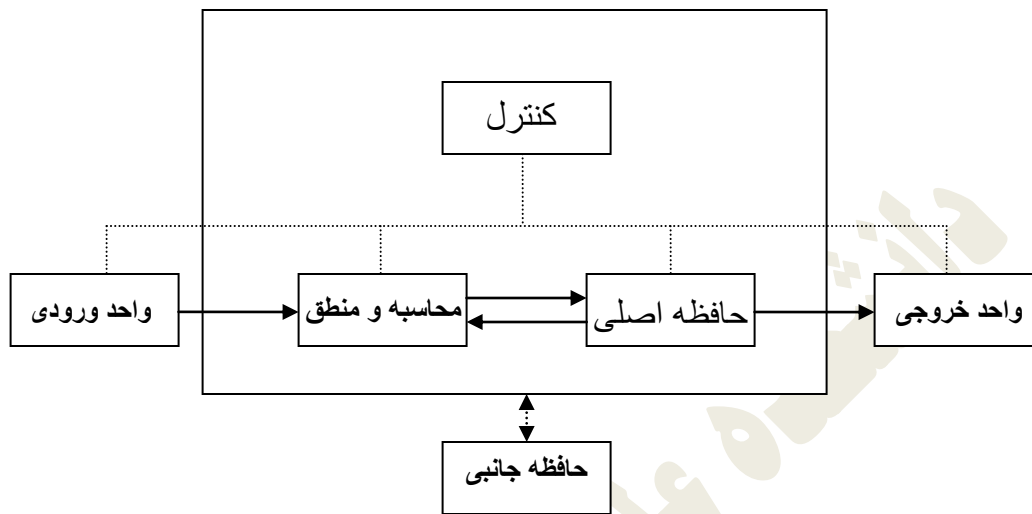
²⁵ touch screen

²⁶ Scanner

²⁷ Output unit

²⁸ Monitor

²⁹ printer



شکل ۲: اجزای تشکیل دهنده یک سیستم کامپیوتری

واحد محاسبه و منطق^{۳۲}: واحدی است که تمامی عملیات ریاضی همچون جمع، ضرب، تفریق، تقسیم و منطقی

همچون مقایسه دو مقدار و ... در آن انجام می پذیرد.

واحد کنترل^{۳۳}: این بخش نقش نظارت و کنترل بر ورود اطلاعات از طریق ورودی، ذخیره آنها در حافظه، انتقال

اطلاعات از حافظه به واحد محاسبه و منطق و خروج اطلاعات از طریق واحد خروجی را دارد. بطور کلی وظیفه کنترل سایر بخشها را بعهده دارد و تصمیم میگیرد کدام عمل در چه زمانی صورت پذیرد و چه مداراتی فعال و یا غیر فعال گردند.

واحد کنترل به همراه واحد محاسبه و منطق بخش مهم تشکیل دهنده واحد پردازش مرکزی یا CPU^{۳۴} هستند. اجزای

دیگر تشکیل دهنده CPU عبارتند از ثباتها^{۳۵} و حافظه نهان یا کش^{۳۶}.

³⁰ plotter

³¹ speaker

³² Arithmetic/logic unit

³³ Control unit

³⁴ Central Processing Unit

³⁵ Register

³⁶ Cache

ثباتها حافظه هایی با حجم بسیار کمی هستند که داده ها برای پردازش و بعد از پردازش در هنگام انتقال به حافظه در ثبات ها قرار می گیرند.

حافظه کش یک نوع حافظه با ظرفیت کم ولی بسیار سریع که بین CPU و حافظه اصلی قرار می گیرد سرعت دستیابی اطلاعات از حافظه کش نسبت به حافظه اصلی بیشتر و نسبت به حافظه های جانبی خیلی بیشتر است. بدین گونه که داده هایی را که CPU از حافظه اصلی فراخوانی می کند پیش از ارسال در کش قرار می گیرد که در فراخوانی های بعدی آن داده ها، با سرعت بیشتری به CPU ارسال شوند.

واحد حافظه^{۳۷} : این واحد وظیفه نگهداری اطلاعات (شامل داده ها و برنامه ها) را بصورت موقت و دائم بر عهده دارد. حافظه ها به دو دسته تقسیم می شوند:

۱. **حافظه اصلی^{۳۸}** : در واقع هر برنامه ای برای اجرا، ابتدا باید به همراه داده های مورد نیاز وارد حافظه اصلی گردد. ویژگی اصلی حافظه اصلی آنست که از سرعت بسیار بالایی برخوردار است اما با قطع برق اطلاعات آن از بین می رود. حافظه اصلی به دو دسته اصلی تقسیم می گردد :

- **حافظه با دستیابی تصادفی^{۳۹}** : این حافظه قابل خواندن و نوشتن می باشد و برای ذخیره اطلاعات کاربران بکار می رود.

- **حافظه فقط خواندنی^{۴۰}** : این حافظه فقط قابل خواندن است و محتویات آن قابل تغییر نیست. این حافظه معمولاً در کارخانه سازنده پر شده و حاوی دستورالعملهای لازم برای راه اندازی اولیه کامپیوتر می باشد.

۲. **حافظه جانبی^{۴۱}** : این حافظه نسبت به حافظه اصلی سرعت کم تری دارد ولی اطلاعات ذخیره شده در آن با قطع برق از بین نمی رود و حجم ذخیره سازی داده در آنها نسبت به حافظه اصلی بسیار زیاد است. حافظه جانبی انواع گوناگونی دارند که می توان CD ، DVD ، هارد دیسک^{۴۲} ، Flash Memory ها ... را نام برد.

³⁷ Memory

³⁸ Main Memory

³⁹ RAM Random Access Memory

⁴⁰ ROM Read Only Memory

⁴¹ Secondary Memory

⁴² Hard disk

در هنگام کار با داده ها در حافظه با اصلاحاتی روبرو می شویم که در زیر به معرفی آنها می پردازیم

بیت^{۴۳}: حافظه از واحدهای کوچکی بنام بیت تشکیل شده است که هر بیت قابلیت نگهداری یک 0 یا 1 را در خود دارد.

بایت^{۴۴}: به هر ۸ بیت یک بایت گفته می شود که واحد اندازه گیری حافظه است.

کاراکتر^{۴۵}: در کامپیوترها کار با اطلاعات بر مبنای بیت دشوار است از این رو از کاراکترها استفاده می شود که بتوان اطلاعات را به صورت ارقام، حروف و نمادها نمایش داد. برای ذخیره سازی کاراکترها به هریک از آنها یک کد عددی نسبت داده شده است و در حقیقت کد عددی هر کاراکتر در کامپیوتر ذخیره می گردد. در گذشته پر کاربردترین کد مورد استفاده، کد ASCII بود که برای نمایش هر کاراکتر از یک بایت استفاده می کرد. از آنجا که هر بایت می تواند بین 0 تا 255 تغییر کند، بنابراین تا ۲۵۶ کاراکتر قابل تعریف است. از این بین کدهای بین 0 تا 127 بصورت استاندارد برای علائم و حروف انگلیسی تعریف شده است و کدهای بالاتر از ۱۲۷ برای هر کشور خالی گذاشته شده است تا بتوانند حروف خاص زبان خود را تعریف کنند. شکل ۳ کدهای ASCII حروف بزرگ انگلیسی را نشان می دهد:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90

شکل ۳

فیلد^{۴۶}: فیلدها از کاراکترها تشکیل شده اند. در واقع گروهی از کاراکترهایی که معنای خاصی دارند. به عنوان مثال

فیلدی که شامل تنها کاراکترهای حروفی باشد می توان آن را فیلد نام یا نام خانوادگی در نظر گرفت.

رکورد^{۴۷}: به مجموعه ای از فیلدهای مرتبط به هم یک رکورد می گوئیم. به عنوان مثال یک رکوردی تحت عنوان

کارمند شامل فیلدهای زیر است:

⁴³ bit
⁴⁴ Byte
⁴⁵ character
⁴⁶ Fields
⁴⁷ Record

- فیلد شماره کارمندی
- فیلد نام کوچک
- فیلد نام خانوادگی
- فیلد سال تولد
- فیلد آدرس
- و

فایل^{۴۸}: به مجموعه ای از رکوردهای مرتبط به هم فایل گفته می شود. به عنوان مثال فایل کارمندان شامل چندین رکورد کارمند است.

همانطور که گفتیم کوچکترین واحد حافظه بیت است. واحد حافظه بزرگتر از بیت، بایت است که از ۸ بیت تشکیل شده است، واحدهای بزرگتری برای سنجش میزان حافظه وجود دارد که در زیر آنها را بیان می کنیم:

کلمه^{۴۹}: معمولاً هر ۲ یا ۴ بایت را یک کلمه در نظر می گیرند

کیلوبایت (KB)^{۵۰}: هر ۱۰۲۴ بایت یک کیلو بایت را تشکیل می دهد. در سیستم دودویی هر کیلو بایت معادل 2^{10} می

باشد واحد های بزرگتر از کیلو بایت عبارتند از

1 MegaByte or 1M = 1024 KiloByte

1 GigaByte or 1G = 1024 MegaByte

1 TeraByte or 1T = 1024 GigaByte

۱،۵- نرم افزار کامپیوتر

سخت افزار به تنهایی توانایی انجام خواسته های کاربر و اجرای برنامه ها را ندارد از این رو برای بکارگیری سخت افزار از

نرم افزار ها استفاده می کنیم. بطور کلی نرم افزار کامپیوتر به دو دسته اصلی تقسیم می گردد :

⁴⁸ File

⁴⁹ word

⁵⁰ Kilo Byte

- نرم افزارهای کاربردی^{۵۱}: نرم افزارهایی هستند که برای یک کاربرد خاص و رفع یک نیاز مشخص کاربران نوشته شده اند. مانند نرم افزار Word، ویرایش عکس و ...
- نرم افزارهای سیستمی: نرم افزارهایی هستند که برای ایجاد و یا اجرای برنامه های کاربردی نوشته می شوند. و به سه گروه تقسیم می شوند

۱. **سیستم عامل**^{۵۲}: سیستم عامل نرم افزاری است که ارتباط بین سخت افزار و کاربران (یا برنامه های کاربردی کاربران) را فراهم می سازد. در حقیقت سیستم عامل مدیریت منابع سخت افزاری یک کامپیوتر را بعهده دارد. چنانچه سیستم عامل نبود، کاربران مجبور بودند مستقیماً و به زبان ماشین با سخت افزار صحبت نمایند که کار مشکلی بود. بهمین دلیل کلیه کاربران مجبورند با یکی از سیستم عاملهای موجود آشنا باشند. در حال حاضر دو سیستم عامل معروف برای کامپیوترهای شخصی وجود دارد: Windows که بیشتر در منازل و محیطهای اداری مورد استفاده قرار می گیرد و Linux که بیشتر در محیطهای دانشگاهی و بعنوان سرویس دهنده استفاده می شود. سیستم عامل Unix نیز بیشتر در کامپیوترهای بزرگ نصب می شود. سیستم عامل Android نیز که امروزه بطور گسترده ای در موبایل ها و تبلت ها مورد استفاده قرار می گیرند.

۲. **برنامه های کمکی**^{۵۳}: این برنامه ها استفاده از کامپیوتر را آسان تر می کند. از طریق این برنامه ها ارتباط کاربر با سخت افزار و برنامه های دیگر آسان تر می شود. از جمله این برنامه ها می توان به برنامه های مدیریت فضای دیسک و ویروس یاب ها اشاره کرد.

۳. **مترجم ها**: همانطور که گفتیم زبان کامپیوتر زبان ماشین است ولی برنامه نویسی به این زبان برای برنامه نویسان مشکل می باشد. برای اینکه کامپیوتر بتواند برنامه های نوشته شده به زبان سطح بالا را درک کند باید از مترجم استفاده کند. از این رو وظیفه مترجم تبدیل دستورات زبان سطح بالا به زبان قابل فهم برای کامپیوتر است. دو نوع اصلی مترجم داریم که عبارتند از:

- **کامپایلر**^{۵۴}: ابتدا کل برنامه زبان سطح بالا را بررسی کرده و در صورت نبود خطا کل آن را به زبان ماشین تبدیل می کند. اکنون برنامه آماده اجرا است.

⁵¹ Application Software

⁵² Operation System

⁵³ Utilitues

• مفسر^{۵۵}: برنامه زبان سطح بالا را دستور به دستور به زبان ماشین تبدیل و همزمان آن را اجرا می کند.

۱,۶- نمایش اطلاعات در کامپیوتر

اطلاعات در کامپیوتر به دو دسته اصلی تقسیم می گردند:

- اطلاعات کاراکتری (حرفی): مانند: A B ... Z \$ # @ !
- اطلاعات عددی که خود به دو دسته اعداد صحیح و اعداد اعشاری تقسیم می گردند.

برای نمایش اطلاعات در کامپیوتر از مبنای ۲ استفاده می گردد .

سیستم اعداد

از کودکی یاد گرفته ایم که برای شمارش از اعداد دهدهی استفاده کنیم و با مفاهیمی مانند یکان، دهگان، صدگان و ... آشنا شده

ایم. درواقع در ریاضیات متداول هر عدد N بصورت زیر تفسیر می گردد :

$$N = (a_{n-1} a_{n-2} \dots a_2 a_1 a_0)_{10} = a_0 \times 10^0 + a_1 \times 10^1 + a_2 \times 10^2 + \dots + a_{n-1} \times 10^{n-1}$$

بعنوان مثال عدد ۶۵۰۹۸ بصورت زیر تفسیر می گردد :

$$(98065)_{10} = 5 \times 10^0 + 6 \times 10^1 + 0 \times 10^2 + 8 \times 10^3 + 9 \times 10^4$$

در سیستم دهدهی می توان از ۱۰ رقم که از مجموعه ارقام {0,1,2,3,4,5,6,7,8,9} تشکیل شده اند، استفاده کرد.

اما می توان اعداد را در هر مبنای دلخواه دیگری مانند b نیز نشان داد در اینصورت هر عدد مانند N در مبنای b بصورت زیر

تفسیر می گردد :

$$N = (a_{n-1} a_{n-2} \dots a_2 a_1 a_0)_b = a_0 \times b^0 + a_1 \times b^1 + a_2 \times b^2 + \dots + a_{n-1} \times b^{n-1}$$

در سیستم عددی بر مبنای b می توان از b رقم که از مجموعه ارقام {0,1,2,3,...,b-2,b-1} تشکیل شده اند، استفاده کرد

⁵⁴ Compiler
⁵⁵ Interpreter

برای مثال نمایش اعداد در مبنای ۷ را در نظر بگیرید. ارقامی که برای نمایش اعداد در این مبنا بکار می رود از مجموعه ارقام $\{0,1,2,3,4,5,6\}$ تشکیل شده اند. با در نظر گرفتن این مبنا نمایش های اعدادی که در این مبنا در ستون سمت چپ جدول آمده اند درست و نمایش اعداد در ستون سمت راست جدول ۱ نادرست است

$$(123456)_7 \quad \checkmark$$

$$(7654321)_7 \quad \times$$

$$(654632)_7 \quad \checkmark$$

$$(5432816)_7 \quad \times$$

جدول ۱

تبدیل مبنایها

برای تبدیل یک عدد از مبنای ۱۰ به هر مبنای دلخواه b ، از روش تقسیمات متوالی استفاده می گردد. بدین ترتیب که عدد مورد نظر بر b تقسیم می گردد و باقیمانده ذخیره می گردد. سپس همین عمل بر روی خارج قسمت تقسیم انجام می شود و عملیات تا زمانی که خارج قسمت به 0 برسد ادامه پیدا می کند. در پایان باقیمانده های ذخیره شده به ترتیب از آخرین باقیمانده تا اولین باقیمانده به ترتیب از چپ به راست نوشته می شوند و عدد حاصل از این فرایند عدد مورد نظر در مبنای b را است.

در زیر روند تبدیل مبنای عدد $(897)_{10}$ به مبنای ۷ نشان داده شده است

$$\begin{array}{r|l}
 897 & 7 \\
 \hline
 896 & 128 \\
 \hline
 \textcircled{1} & 126 \\
 \hline
 & \textcircled{2} \quad 14 \\
 & \hline
 & \textcircled{4} \quad 0 \\
 & \hline
 & \textcircled{2} \quad 0 \\
 & \hline
 & 0
 \end{array}$$

$$(897)_{10} = (2421)_7$$

برای تبدیل از مبنای دیگر به مبنای ۱۰ باید اعداد را در ارزش مکانی خود ضرب کنیم و حاصل ضرب ها را با هم جمع کنیم. عدد حاصل آن عدد در مبنای ۱۰ است. برای مثال بالا داریم

$$(2421)_7 = 2 \times 7^3 + 4 \times 7^2 + 2 \times 7^1 + 1 \times 7^0 = 686 + 196 + 14 + 1 = 897$$

از این رو تبدیل از مبنای ۲ به مبنای ۸ به سادگی انجام می گیرد، ابتدا ارقام را از راست به چپ بصورت دسته های ۳ تایی تقسیم کنید و سپس برای هر دسته معادل آن را در مبنای ۸ قرار دهید. و برای دسته آخر اگر تعدادشان کمتر از ۳ بود به سمت چپ آن ۰ اضافه می کنیم تا تعدادشان به ۳ برسد..

. در زیر تبدیل عدد $(1011111)_2$ به مبنای ۸ نشان داده شده است.

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline & & & & 2 & 3 & & & 7 & \end{array} = (237)_8$$

همانطور که مشاهده می شود نمایش یک عدد در مبنای هشت نسبت به نمایش همان عدد در مبنای دو ارقام کمتری دارد. مبنای شانزده یکی دیگر از مبنایها است که بسیار مورد استفاده قرار می گیرد. مبنای شانزده همانند مبنای هشت است با این تفاوت که هر ۴ رقم در مبنای دو یک عدد در مبنای شانزده است. مشکلی که در نمایش عدد در مبنای شانزده وجود دارد این است که در این مبنای نیاز به ۱۶ رقم داریم در حالیکه ارقام موجود فقط ۱۰ تا است. بهمین دلیل از حروف A تا F برای ارقام ۱۰ تا ۱۵ استفاده می گردد. یعنی ارقام عبارتند از :

0 1 2 3 4 5 6 7 8 9 A B C D E F

مبنای ۱۶	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹	A	B	C	D	E	F
مبنای ۲	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

برای تبدیل اعداد از مبنای ۲ به ۱۶ از همان روش گفته شده برای مبنای ۸ استفاده می نمایم با این تفاوت که از سمت راست بصورت دسته های ۴ تایی جدا می کنیم و برای دسته آخر اگر تعدادشان کمتر از ۴ بود به سمت چپ آن ۰ اضافه می کنیم تا تعدادشان به ۴ برسد. در زیر تبدیل عدد $(1010011101)_2$ به مبنای ۱۶ را نشان می دهد

$$\begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ \hline & & 2 & & 9 & & & & D & & & \end{array} = (29D)_{16}$$

همانطور که مشاهده می کنید یک عدد در مبنای شانزده نیز نسبت به نمایش همان عدد در مبنای دو ارقام کمتری دارد.

برای تبدیل از مبنای ۸ به مبنای ۲ بازای هر رقم مبنای ۸، ۳ رقم در مبنای ۲ قرار می دهیم و برای تبدیل از مبنای ۱۶ به مبنای ۲ بازای هر رقم مبنای ۱۶، ۴ رقم در مبنای ۲ قرار می دهیم.

در مثال زیر عدد $(367)_8$ و عدد $(3BF)_{16}$ را به مبنای ۲ تبدیل می کنیم.

$$(367)_8 = (11110111)_2$$

$\begin{array}{ccc} \swarrow & \downarrow & \searrow \\ 011 & 110 & 111 \end{array}$

$$(3BF)_{16} = (1110111111)_2$$

$\begin{array}{ccc} \swarrow & \downarrow & \searrow \\ 0011 & 1011 & 1111 \end{array}$

برای تبدیل یک عدد اعشاری از مبنای ۱۰ به مبنای ۲ نیز کافی است عدد را به صورت مجموع توانهای منفی عدد ۲ نوشت به عنوان مثال داریم:

$$(0.75)_{10} = (0.11)_2 = 1 \times 2^{-1} + 1 \times 2^{-2} = 0.5 + 0.25$$

از قاعده زیر استفاده میکنیم: برای تبدیل عدد

مراحل زیر را انجام بده $N \neq 0$ - تا زمانی که

۱، ۱ - قسمت صحیح $N = N \times 2$ int

۱، ۲ - قسمت اعشاری $N = N \times 2$ float

۱، ۳ - چاپ گن int N

۱، ۴ - $N = \text{float } N$

برای تبدیل یک عدد اعشاری از مبنای ۲ به مبنای ۱۰ کافی است، هر رقم را در توانی از ۲ ضرب نمود، که اولین توان ۲ بعد از ممیز 1- شروع و برای یک عدد اعشاری n رقمی به 2^{-n} ختم میشود.

مثلا:

$$(0.101)_2 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 0.5 + 0.25 + 0.125 = 0.875$$

۱،۷- کد اسکی

در سال ۱۹۶۸ موسسه ملی استاندارد آمریکا یک کد ۷بیتی برای تمام حروف الفبا، ارقام ۰ تا ۹، نمادهای نقطه گذاری که در اغلب ماشین های تحریر به کار میروند و چند کد کنترل خاص پایه گذاری کرد. آنها این کد را کد اسکی (ASCII) نامیدند.

گرچه کد اسکی یک کد ۷بیتی است ولی غالبا به صورت یک بایت (۸بیت) نوشته شده از بیت هشتم آن یا صرفنظر میشود یا برای توازن استفاده میشود. فرضا در این کدگذاری کد حرف بزرگ L برابر با 0100,1100 یا 4C در مبنای ۱۶ است. امروزه تقریبا متنها با فرمت کد اسکی کد میشوند، به این ترتیب انتقال اطلاعات بین دو سیستم کامپیوتری مختلف ممکن میشود. هر بار که یکی از کلیدهای صفحه کامپیوتر را فشار دهید یک کد اسکی برای پردازش به کامپیوتر فرستاده میشود.

لازم به ذکر است که علاوه بر کداسکی کدهای دیگری نیز وجود دارد مانند کد گری و کد BCD.

تمرینها

۱- اعداد زیر را به مبناهای خواسته شده تبدیل نمایید.

$$(1234)_{10} = (?)_6$$

$$(1030)_{10} = (?)_2$$

$$(1010101010101)_2 = (?)_8 = (?)_{16}$$

$$(ABCF)_{16} = (?)_{10} = (?)_8$$

$$(654)_8 = (?)_{10} = (?)_2 = (?)_{16}$$

۲- نمایش دهدهی ۲۵۰/۵ در مبنای ۳ چیست؟

۲.۱- الگوریتم

تعریف عامیانه ای که می شود برای الگوریتم ارائه کرد روال خاص انجام کاری است. الگوریتم در لغت به معنای روش حل مسائل می باشد و از نام دانشمند و ریاضی دان ایرانی ، ابو جعفر بن محمد خوارزمی گرفته شده است.

برای اینکه کامپیوترها بتوانند یک کار را بطور کامل انجام دهند باید تمام مراحل انجام آن کار بطور دقیق برایشان مشخص شده باشد.

تعریف دقیقی که می شود برای الگوریتم ارائه کرد این است که: " به مجموعه ای از دستورالعمل ها که مراحل مختلف (برای حل یک مسئله) را به زبان قطعی و با جزئیات کافی بیان کرده و در آن ترتیب مراحل و خاتمه پذیر بودن عملیات در آن کاملا مشخص باشد را الگوریتم می نامند."

در هنگام نوشتن هر الگوریتم باید به نکات زیر توجه کرد :

- ورودی : یک الگوریتم می تواند صفر یا چند ورودی داشته باشد که از محیط خارج تامین می گردد.
- خروجی : الگوریتم باید یک یا چند کمیت خروجی داشته باشد.
- زبان دقیق: اگر از یک جمله برداشتهای متفاوتی شود و یا اینکه جمله مبهم باشد ، گوییم بیان آن جمله یا عبارت دقیق نیست.
- جزئیات کافی: عملیات ذکر شده در یک الگوریتم باید برای مجری آن شناخته شده باشد. نویسنده الگوریتم باید از چیزهایی که مجری الگوریتم (زبان برنامه نویسی مورد نظر) می داند آگاه باشد و هیچگونه فرضی را در مورد مجری الگوریتم منظور نکند.

⁵⁶ Algorithm

⁵⁷ Flowchart

- ترتیب مراحل : مراحل انجام دستورالعملها باید به ترتیب انجام گیرد و در غیر اینصورت دستورالعملها قابل اجرا نخواهد بود.

- کارایی : هر دستورالعمل باید قابل اجرا باشد.

- خاتمه عملیات: شرط یا شروط خاتمه عملیات باید در الگوریتم ذکر شود بخصوص وقتی که عملیات تکراری هستند

معمولا برای حل یک مسئله، مراحل زیر طی می گردد:

- تعریف مسئله بصورت جامع و دقیق (شامل تعریف ورودیها و خروجیها)

- بررسی راه حل‌های مختلف برای حل مسئله

- انتخاب مناسبترین راه حل و تهیه یک الگوریتم برای آن

- آزمایش الگوریتم با داده های ورودی و اشکالزدایی آن

- تبدیل الگوریتم به یک زبان برنامه نویسی کامپیوتری

- وارد کردن برنامه به کامپیوتر و تست و اشکالزدایی آن

- استفاده از برنامه

برای بیان الگوریتم ها از زبان طبیعی استفاده می کنیم. گاهی در قسمت هایی از الگوریتم از دستورات زبان برنامه نویسی خاصی استفاده می شود که به آنها شبه کد گفته می شود.

۲،۲-قرارداد کار با الگوریتم ها

در زیر قراردادهایی وجود دارند که در هنگام نوشتن الگوریتم باید به آن توجه کرد:

- در ابتدای هر الگوریتم کلمه شروع و در انتهای آن از کلمه پایان استفاده می کنیم.

- برای هر یک از دستورالعمل ها شماره ای در نظر می گیریم.

- برای دریافت اطلاعات از کاربر از دستور بخوان استفاده می گردد.

- برای نوشتن اطلاعات در خروجی از دستور چاپ کن استفاده می گردد.

- برای انجام محاسبات ریاضی یا کار بر روی داده ها از داده مکانی برای ذخیره داده ها و اضافه کردن نتایج در نظر می گیریم که در کامپیوتر این مکان ها در خانه های حافظه ذخیره می شوند. برای اینکه بتوانیم بعدا به این خانه های حافظه مراجعه کنیم، به هریک از آنها یک نام نسبت می دهیم. به هریک از این نامها یک متغیر گفته می شود. دلیل این نامگذاری آنستکه مقادیر ذخیره شده در هریک از این خانه های حافظه می تواند تغییر کند. نام متغیرها بصورت دلخواه در نظر گرفته می شود ولی بهتر است نام آن متناسب با کاربرد باشد.
- برای انتساب یک مقدار به یک متغیر از علامت \leftarrow استفاده می شود.

مثال 2.1: الگوریتمی بنویسید که ۳ عدد را از کاربر دریافت کند و میانگین آنها را چاپ کند.

۱. شروع
۲. A, B, C را بخوان
۳. $S \leftarrow A+B+C$
۴. $Average \leftarrow S/3$
۵. $Average$ را چاپ کن
۶. پایان

۲,۳- جملات شرطی

گونه ای از جملاتی که در هنگام الگوریتم نویسی از آنها استفاده می کنیم جملات شرطی هستند که شرط یا شرایطی خاصی را چک می کنند که در صورت برقراری آن عملیات خاصی را انجام دهند و در صورت عدم برقراری عملیات دیگری را انجام دهند. از این رو دو نوع جمله شرطی داریم:

- شرطی نوع ساده
- شرطی نوع دو

۲,۴- شرطی نوع ساده:

شکل کلی دستورات شرطی نوع ساده بصورت زیر است:

اگر (عبارت شرطی) آنگاه دستورات

و بدین صورت تعبیر می شود که اگر عبارت شرطی درست باشد آنگاه قسمت دستورات اجرا خواهد شد. اما در غیر اینصورت به دستور بعدی خواهد رفت.

۲,۵- شرطی نوع دو:

شکل کلی این نوع شرط به صورت زیر است:

اگر (عبارت شرطی) آنگاه دستورات ۱

در غیر اینصورت دستورات ۲

اگر عبارت شرطی درست باشد دستورات ۱ اجرا می شود و اگر درست نباشد دستورات ۲ اجرا خواهند شد.

مثال ۲,۱: الگوریتمی بنویسید که عددی را دریافت کند و اگر عدد زوج بود کلمه زوج و اگر فرد بود کلمه فرد را در خروجی چاپ

کند

۱. شروع کن

۲. عدد A را دریافت کن

۳. $B \leftarrow A/2$

۴. $C \leftarrow A-2*B$

۵. اگر C برابر با صفر بود آنگاه در خروجی زوج را چاپ کن در غیر اینصورت فرد

۶. پایان

مثال ۲,۳: الگوریتمی بنویسید که ضرایب یک معادله درجه دوم بصورت $x^2 + b x + c = 0$ را دریافت و ریشه های آن را در

صورت وجود محاسبه و چاپ کند.

۱. شروع

۲. a و b و c را بخوان

۳. اگر $(a=0)$ آنگاه چاپ کن "معادله درجه دو نیست" و به مرحله پایان برو.

۴. $\text{delta} \leftarrow b^2 - 4ac$

۵. اگر $(\text{delta} < 0)$ آنگاه چاپ کن "معادله جواب ندارد"

در غیر اینصورت $x_1 \leftarrow \frac{-b - \sqrt{\text{delta}}}{2a}$ و $x_2 \leftarrow \frac{-b + \sqrt{\text{delta}}}{2a}$ و x_1 و x_2 را چاپ کن

۶. پایان

مثال ۲,۴: الگوریتمی بنویسید که ۳ عدد را دریافت کند و مشخص کند که می توان با این ۳ عدد مثلث ساخت یا خیر

همانطور که می دانید شرط اینکه سه عدد a, b و c تشکیل یک مثلث بدهند این است که روابط زیر برقرار باشد:

$$\begin{cases} a \leq b + c \\ b \leq a + c \\ c \leq b + a \end{cases}$$

۱. شروع

۲. a, b و c را بخوان

۳. اگر $a \leq b + c$ آنگاه به خط ۴ برو در غیر اینصورت به خط ۷ برو

۴. اگر $b \leq a + c$ آنگاه به خط ۴ برو در غیر اینصورت به خط ۷ برو

۵. اگر $c \leq b + a$ آنگاه به خط ۴ برو در غیر اینصورت به خط ۷ برو

۶. چاپ کن می توان یک مثلث ساخت

۷. چاپ کن نمی توان یک مثلث ساخت

۸. توقف

۲,۵- حلقه های تکرار

برای حل مسائل گاهی اوقات لازم است که عملیات مشخصی چندین بار تکرار شوند. به عنوان مثال برای نوشتن الگوریتمی که ۵۰ عدد را از کاربر دریافت کند تا میانگین آنها را حساب کند نباید ۵۰ بار "دستور عدد را بخوان" و "آن را با عددهای خوانده شده جمع کن" را نوشت. راه حل بهتر آنست که بگونه ای به الگوریتم بگوییم بنحوی عمل خواندن اعداد و جمع زدن آنها را ۵۰ بار تکرار کند.

حلقه تکرار بگونه ای است که مجموعه ای از دستورات را تا زمانیکه شرط خاصی برقرار باشد تکرار می کند.

بطور کلی حلقه های تکرار به دو شکل مورد استفاده قرار می گیرد :

- حلقه های شرطی که شرط آنها در ابتدای حلقه چک می شوند و دارای ساختار زیر هستند:

تا زمانیکه (شرط مورد نظر) دستورات a تا b را تکرار کن

...(a

.

.

.

... (b

در این حالت ابتدا شرط مورد نظر بررسی می گردد؛ در صورتیکه شرط برقرار نباشد به اولین دستور پس از b می رود. اما در صورتیکه شرط درست ارزیابی شود، دستورات شماره a تا b انجام می شوند و سپس مجدداً به ابتدای حلقه بازگشته و عملیات فوق را مجدداً تکرار می کند.

- حلقه های شرطی که شرط در انتهای حلقه چک می شود و دارای ساختار زیر هستند :

تکرار کن

...(a

... (b)

تا زمانیکه (شرط مورد نظر)

در این روش ابتدا دستورات حلقه یکبار انجام می شوند و در پایان حلقه شرط بررسی می گردد. چنانچه شرط برقرار نبود به دستور بعدی می رود و در صورت برقرار بودن شرط، مجدداً به ابتدای حلقه باز می گردد.

مثال ۵: الگوریتمی بنویسید که n عدد را به دلخواه از ورودی دریافت کند و حاصلجمع آنها را چاپ نماید.

برای حل این مسئله از متغیر i برای شمارش تعداد ورودی های دریافت شده استفاده می کنیم. از متغیر Sum برای ذخیره حاصلجمع اعداد خوانده شده استفاده می کنیم. از متغیر A برای دریافت اعداد استفاده می کنیم.

۱. شروع

۲. عدد n را بخوان

۳. $Sum \leftarrow 0$

۴. $i \leftarrow 1$

۵. تا زمانیکه $i \leq n$ است دستورات ۶ تا ۸ را تکرار کن

۶. A را بخوان

۷. $Sum \leftarrow Sum + A$

۸. $i \leftarrow i + 1$

۹. Sum را چاپ کن

۱۰. توقف

مثال ۶,۲: الگوریتمی بنویسید که یک عدد را دریافت و فاکتوریال آن را محاسبه و چاپ کند.

$$N! = 1 \times 2 \times 3 \times \dots \times (N-1) \times N$$

عدد ورودی برای محاسبه فاکتوریال آن : n

شمارنده حلقه : i

مقدار فاکتوریال : $fact$

(۱) شروع

(۲) n را بخوان

(۳) $i \leftarrow 1$ و $fact \leftarrow 1$

(۴) تا زمانی که ($i \leq n$) دستورات 5-6 را تکرار کن

(۵) $fact \leftarrow fact \times i$

(۶) $i \leftarrow i + 1$

(۷) $fact$ را چاپ کن

(۸) توقف کن

مثال ۲,۷: الگوریتمی بنویسید که n عدد صحیح را دریافت و حداکثر و حداقل آنها را چاپ کند.

برای حل مسائل حداکثر یا حداقل ابتدا باید یک متغیر برای نگهداری آنها در نظر بگیریم. نکته مهم مقدار اولیه این متغیر است. دو روش کلی وجود دارد:

○ اولین مقدار را دریافت و آن را بعنوان مقدار اولیه در نظر بگیریم.

○ مقدار اولیه متغیر حداکثر را برابر کمترین عدد ممکن و متغیر حداقل را برابر بیشترین عدد ممکن در نظر

بگیریم.

عدد دریافت شده در هر مرحله : A شمارنده حلقه : i تعداد اعداد : n

بزرگترین عدد صحیح : min کوچکترین عدد صحیح : max

(۱) شروع

(۲) n را بخوان

(۳) A را بخوان

(۴) $min \leftarrow A$ و $max \leftarrow A$

(۵) $i \leftarrow 2$

(۶) تا زمانی که ($i \leq n$) دستورات ۷ تا ۱۰ را تکرار کن

(۷) A را بخوان

(۸) اگر ($A > max$) آنگاه $max \leftarrow A$

۹) در غیر اینصورت اگر ($A < \min$) آنگاه $\min \leftarrow A$

۱۰) $i \leftarrow i + 1$

۱۱) \min و \max را چاپ کن

۱۲) توقف کن

مثال ۲,۸: الگوریتمی بنویسید که عدد طبیعی n را دریافت کند و مجموع عبارات زیر را محاسبه و چاپ نماید:

$$s = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

۱. شروع

۲. N را بخوان

۳. $S \leftarrow 1$

۴. $i \leftarrow 2$

۵. تا زمانی که ($i \leq n$) دستورات 6 تا 7 را تکرار کن

۶. $s \leftarrow s + \frac{1}{i}$

۷. $i \leftarrow i + 1$

۸. s را چاپ کن

۹. توقف

$$s = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

۱. شروع

۲. N را بخوان

۳. $s \leftarrow 0$

۴. $i \leftarrow 1, k \leftarrow 1, P \leftarrow 1$

۵. تا زمانی که ($i \leq n$) دستورات 6 تا 9 را تکرار کن

۶. $s \leftarrow s + \frac{k}{P}$

$$k \leftarrow -k \quad .7$$

$$i \leftarrow 1+i \quad .8$$

$$P \leftarrow P+2 \quad .9$$

۱۰. s را چاپ کن

۱۱. توقف

مثال ۲,۹: الگوریتمی بنویسید که یک عدد را دریافت و وارون آن را محاسبه و به همراه خود عدد چاپ کند. مثلا وارون عدد ۱۳۹۲ برابر ۲۹۳۱ می باشد.

برای وارون کردن هر عدد ابتدا باید ارقام آن را از سمت راست جدا کنیم. ساده ترین راه برای اینکار، بدست آوردن باقیمانده تقسیم عدد بر ۱۰ است. مثلا $1392 \bmod 10 = 2$ فرض می کنیم عملگر mod بمعنای باقیمانده قبلا تعریف شده است، در غیر این صورت برای بدست آوردن باقیمانده عددی مانند a به شکل روبرو عمل می کنیم $a \bmod 10 = a - ((a/10) * 10)$ تنها نکته آنستکه این فرمول برای اعداد منفی کار نمی کند. لذا برای اعداد منفی ابتدا باید آنها را مثبت کرده و در پایان مجددا تبدیل به عدد منفی می کنیم.

متغیر کمکی برای انجام عملیات بر روی عدد مورد نظر : a عدد مورد نظر : adad

وارون عدد مورد نظر : varoon مقدار باقیمانده تقسیم عدد بر ۱۰ در هر مرحله : remain

۱. شروع

۲. adad را بخوان

۳. اگر (adad < 0) آنگاه $a \leftarrow -adad$ در غیر این صورت $a \leftarrow adad$

۴. $varoon \leftarrow 0$

۵. تا زمانی که (a > 0) آنگاه دستورات ۶ تا ۸ را تکرار کن

۶. $remain \leftarrow a \bmod 10$

۷. $a \leftarrow a / 10$

۸. $varoon \leftarrow varoon * 10 + remain$

۹. اگر (adad < 0) آنگاه $varoon \leftarrow -varoon$

۱۰. adad و varoon را چاپ کن

۱۱. توقف کن

مثال ۲,۱۰: الگوریتمی بنویسید که یک عدد صحیح مثبت در مبنای ۱۰ را دریافت و سپس آن را به مبنای b ببرد. مبنای b نیز از کاربر دریافت می گردد.

مبنای موردنظر : b عدد موردنظر : $adad$

حاصل تبدیل عدد به مبنای موردنظر : $mabnaieb$

باقیمانده در هر مرحله : $remain$ شمارنده حلقه : i

۱. شروع

۲. $adad$ و b را بخوان

۳. $mabnaieb \leftarrow 0$ و $i \leftarrow 0$

۴. تا زمانی که $(adad > 0)$ دستورات 5 تا 8 را تکرار کن

۵. $remain \leftarrow adad \bmod b$

۶. $mabnaieb \leftarrow mabnaieb + remain \times 10^i$

۷. $adad \leftarrow adad / b$

۸. $i \leftarrow i + 1$

۹. $mabnaieb$ را چاپ کن

۱۰. توقف کن

مثال ۲,۱۱: الگوریتمی بنویسید که برای دانشجویان ورودی ۹۱ دانشکده ریاضی، نام و شماره دانشجویی و تعداد دروس را دریافت کند و سپس برای هر دانشجو نمره و تعداد واحد هر درس وی را دریافت و معدل وی را محاسبه نماید. در پایان میانگین معدل دانشکده و بیشترین و کم ترین معدل دانشکده را به همراه شماره دانشجویی آنها چاپ کند.

برای حل این مسئله باید ۲ حلقه متداخل یا تودرتو در نظر بگیریم. یک حلقه بیرونی که تعداد دانشجویان ورودی ۹۰ را می شمارد و یک حلقه داخلی که برای هر دانشجو شماره دانشجویی و تعداد دروس و تعداد واحد درس ها و معدل آن را حساب کند.

نکته مهم آنست که هنگامیکه دو حلقه متداخل داریم، باید از دو متغیر جداگانه برای شمارش هریک استفاده کنیم تا مشکلی پیش نیاید.

تعداد دروس هر دانشجو: dars شمارنده دانشجویان: i تعداد دانشجویان ورودی ۹۰: n
 معدل کل دانشجویان: total Ave تعداد واحد هر درس: vahed شمارنده دروس: j
 تعداد کل واحدهای هر دانشجو: totalVahed کمترین معدل: minAve بیشترین معدل: maxAve
 شماره دانشجویی بیشترین معدل: maxStudentID معدل هر دانشجو: moadel جمع نمره هر دانشجو: Sum
 شماره دانشجویی: StudentID نمره درس: nomre شماره دانشجویی کمترین معدل: minStudentID

- ۱ شروع
- ۲ n را دریافت کن
- ۳ $i \leftarrow 1$
- ۴ $total\ Ave \leftarrow 0$ ، $max\ Ave \leftarrow -1$ ، $min\ Ave \leftarrow 1$ ، $max\ StudentID \leftarrow 0$ ، $min\ StudentID \leftarrow 0$
- ۵ تا زمانیکه $i \leq n$ دستورات ۶ تا ۱۷ را تکرار کن
- ۶ $Sum \leftarrow 0$ ، $moadel \leftarrow 0$ ، $totalVahed \leftarrow 0$
- ۷ dars و StudentID را دریافت کن
- ۸ $j \leftarrow 1$
- ۹ تا زمانیکه $j \leq dars$ دستورات ۱۰ تا ۱۳ را تکرار کن
- ۱۰ Vahed و nomre را دریافت کن
- ۱۱ $sum \leftarrow Vahed \times nomre + sum$
- ۱۲ $totalVahed \leftarrow totalVahed + vahed$
- ۱۳ $j \leftarrow j + 1$
- ۱۴ $moadel \leftarrow sum / totalVahed$
- ۱۵ $totalAve \leftarrow totalAve + moadel$
- ۱۶ اگر $maxAve < moadel$ باشد آنگاه $maxAve \leftarrow moadel$ و $maxStudentID \leftarrow StudentID$ در غیر اینصورت اگر $moadel < minAve$ باشد آنگاه $moadel \leftarrow minAve$ و $minStudentID \leftarrow StudentID$

۱۷ $i \leftarrow i+1$

۱۸ $totalAve \leftarrow totalAve/n$

۱۹ $totalAve$ رو چاپ کن.

۲۰ $maxAve$ رو به همراه $maxStudentID$ چاپ کن

۲۱ $minAve$ رو به همراه $minStudentID$ چاپ کن

۲۲ توقف

مثال ۲,۱۲: الگوریتمی بنویسید که یک عدد را دریافت و تعیین کند که آیا اول است یا خیر؟

می دانیم که عددی اول است که بر هیچ یک از اعداد کوچکتر از خود بجز ۱ بخشپذیر نباشد. یک نگاه دقیقتر نشان می دهد که اگر یک عدد بر کلیه اعداد از ۲ تا جذر خود بخشپذیر نباشد، اول است. الگوریتم زیر بر همین اساس نوشته شده است. دقت کنید که الگوریتم بگونه ای است که به محض اینکه عدد مورد نظر بر یک عدد بخشپذیر باشد، از حلقه خارج شده و اعلام می کند که عدد مورد نظر اول نیست.

نکته : همانگونه که می بینید در شرط حلقه از and استفاده شده است. عبارت دیگر از یک شرط ترکیبی استفاده شده است. در شرطهای ترکیبی می توان از عملگرهای and و or استفاده کرد و معنای آنها نیز به شرح زیر است :

عبارت در صورتی درست است که هر دو شرط درست باشند : C_1 and C_2

عبارت در صورتی درست است که حداقل یکی از دو شرط درست باشد : C_1 or C_2

شمارنده حلقه : i ریشه عدد مورد نظر : $root$ عدد مورد نظر : $adad$

یک سویچ که نشاندهنده اول بودن (۱) یا اول نبودن (۰) عدد است : $primeSw$

۱- شروع

۲- $adad$ را بخوان

$$root \leftarrow \lfloor \sqrt{adad} \rfloor \quad -3$$

$$primeSw \leftarrow 1 \text{ و } i \leftarrow 2 \quad -4$$

-5 تازمانیکه ($i \leq root$ and $primeSw=1$) دستورات ۶ تا ۷ را تکرار کن

$$primeSw \leftarrow 0 \text{ اگر } (adad \bmod i = 0) \text{ آنگاه } \quad -6$$

$$i \leftarrow i + 1 \quad -7$$

-8 اگر ($primeSw = 1$) آنگاه چاپ کن "عدد اول است"

درغیراینصورت چاپ کن "عدد اول نیست"

-9 توقف کن

نکته مهم : پس از نوشتن الگوریتم باید الگوریتم را با داده هایی که کلیه حالات ممکن را پوشش می دهند چک کرد چراکه ممکن است الگوریتم بازای یک ورودی خاص دچار مشکل شود!

۲,۶-فلوچارت:

در گذشته برای درک بهتر الگوریتم ها و سهولت دنبال کردن دستورالعملهای آن از یکسری اشکال خاص برای نشان دادن روند الگوریتم ها استفاده می کردند که به آن فلوچارت گفته می شود. در واقع فلوچارت مجموعه ای از علائم ساده است که الگوریتم را به صورت نمادهای تصویری نمایش می دهد.

۲,۷-علائم فلوچارت

علامت شروع و پایان

در فلوچارت از شکل بیضی برای نمایش شروع و پایان الگوریتم استفاده می کنیم

پایان

شروع

علامت جایگزینی و انتساب

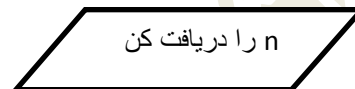
در فلوچارت برای انجام عمل جایگزینی و یا انتساب و یا عمل محاسباتی از مستطیل استفاده می شود.

total ← total/n

i ← 1

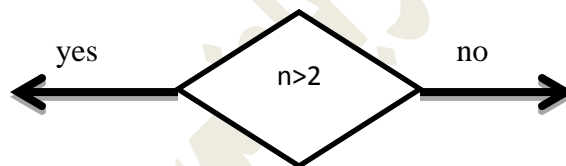
علامت ورودی

در فلوجارت از علامت متوازی الاضلاع برای گرفتن ورودی ها استفاده می شود.



علامت شرط

در فلوجارت از علامت لوزی برای شرط استفاده می شود

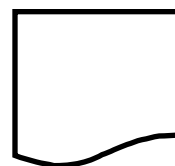


علامت چاپ

در فلوجارت از علائم زیر برای چاپ استفاده می شود

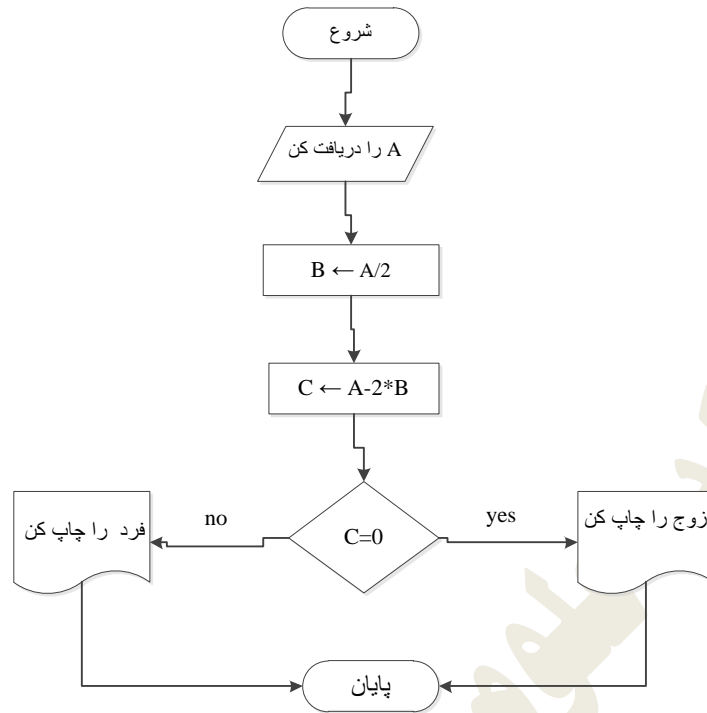


چاپ روی صفحه نمایشگر



چاپ روی کاغذ

از فلش های جهت دار به منظور ارتباط بین علائم استفاده می شود. استفاده از فلوجارت تنها برای الگوریتمهای کوچک مناسب است. از این رو ما تنها با استفاده از مثال ۲ این بخش نحوی استفاده از آن را نمایش می دهیم.



تمرینها

۱- الگوریتمی بنویسید که زمان را بر حسب ثانیه دریافت کند و مشخص کند که از چند ساعت و چند دقیقه و چند ثانیه تشکیل شده است.

۲- الگوریتمی بنویسید که سه عدد a ، b و c را که اضلاع یک مثلث هستند را دریافت و مشخص کند که می توان با آن یک مثلث قائم الزاویه تشکیل داد.

۳- الگوریتمی برای بنویسید که عددی را دریافت کن و مشخص کند که آن عدد کامل هست یا نه؟ (عدد کامل، یک عدد صحیح مثبت است که برابر با مجموع مقسوم علیه های سره مثبت خود (همه مقسوم علیه های مثبتش غیر از خود عدد) باشد).

۴- الگوریتمی بنویسید که یک عدد در مبنای b را دریافت و آن را به مبنای ۱۰ ببرد.

۵- الگوریتمی بنویسید که یک عدد طبیعی را از ورودی دریافت و توان آن را به روش زیر محاسبه کند

توان دوم عدد n = مجموع n عدد فرد از ۱

$$6^2 = 1 + 3 + 5 + 7 + 9 + 11 = 36$$

۶- الگوریتمی بنویسید که ۵۰ جمله از دنباله زیر موسوم به دنباله فیبوناچی را چاپ کند.

1,1,2,3,5,8,...

۷- الگوریتمی بنویسید که عدد i را دریافت نموده و مجموع اعداد سطر i ام مثلث زیر را بیابد

			1					
			1	1				
			1	2	1			
			1	3	3	1		
			1	4	6	4	1	
			1	5	10	10	5	1
		

۸- الگوریتمی بنویسید که عدد n را دریافت نموده و مجموع n جمله از جملات سری زیر را بیابد

$$s = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} + \dots$$