

به نام خالق زیبایی ها

مبانی برنامه سازی و ویژوال بیسیک

ویژه دانشجویان مقطع کاردانی - کامپیوتر

مدرس : سیده شیدا افتخارحسینی

۲- الگوریتم و فلوچارت

• شناخت مسئله

- بررسی داده ها و یا معلومات (ورودیها)، مجهولات (خروجیها) و یافتن ارتباط منطقی بین داده ها و مجهولات
- مثال : یافتن مساحت یک مثلث با داشتن اندازه قاعده و ارتفاع آن
 - داده ها (ورودی) - اندازه ارتفاع و قاعده مثلث
 - مجهولات (خروجی) - مساحت مثلث
 - رابطه منطقی : روش محاسبه مساحت مثلث (ارتفاع \times قاعده \times ۱/۲)
- اغلب، مسائل دارای راه حل‌های گوناگونی می باشند، یافتن بهترین راه حل به ابتکار، تمرین و از همه مهمتر تجربه بستگی دارد.

تعریف الگوریتم

- مجموعه دستورالعملهایی که مراحل حل یک مسئله مشخص کند با ویژگیهای :
 - با یک زبان واضح، روشن و بدون ابهام و پیچیدگی
 - با جزئیات کافی (از دید اجرا کننده الگوریتم)
 - شروع عملیات
 - ترتیب اجرای دستورات
 - پایان عملیات

مجری الگوریتم

- الگوریتم ها می توانند به دو صورت انجام شوند

- توسط ماشین (کامپیوتر)

- بعد از تبدیل شدن به زبان مناسب برای ماشین بطور اتوماتیک توسط ماشین قابل اجرا هستند.

- توسط انسان

- برای حصول اطمینان از عملکرد صحیح الگوریتم، گاهی خودمان آن را بصورت دستی دنبال نموده و در واقع خود مجری الگوریتم می شویم.

کاربرد الگوریتم

- همه ما در طی روز برای انجام کارهای روزمره از روش الگوریتمی (و یا منطقی) استفاده می کنیم.
 - مانند مطالعه کتاب
 - تعویض چرخ پنجر شده
 - پختن غذا
 - پختن کیک و
- در واقع برای انجام هر یک از این کارها، لازم است تعدادی دستورالعملهای ساده تر را به ترتیب مناسب اجراء کرده تا به نتیجه مطلوب برسیم.

مثالی از کاربرد الگوریتم

• کتابی داریم و می خواهیم آنرا مطالعه نماییم، برای مطالعه کتاب از ابتدا تا انتها باید مراحل زیر را انجام دهیم:

- . شروع
- . باز کردن کتاب
- . از خط اول شروع به خواندن می کنیم
- . آیا به انتهای صفحه رسیده ایم یا خیر؟
- . اگر به انتهای صفحه رسیده ایم به مرحله بعدی و در غیر اینصورت به مرحله ۹ می رویم
- . آیا تا انتهای کتاب خوانده شده است یا خیر؟
- . اگر به انتهای کتاب رسیده باشیم به مرحله ۱۰ می رویم
- . صفحه بعدی را باز کرده و به مرحله ۳ می رویم
- . خواندن را ادامه می دهیم و به مرحله ۴ می رویم
- . پایان

چند نکته

- دستورات بیان شده در الگوریتم بایستی برای مجری الگوریتم قابل درک باشد.
- بطور مثال، هنگام نوشتن نسخه توسط پزشک، بیان می شود که مثلا از این شربت ۱ قاشق مرباخوری خورده شود (و نه اینکه مثلا ۲,۵ سی سی میل شود- که برای ما که مجری الگوریتم نسخه پزشک هستیم زیاد قابل فهم نیست!)
- هنگام بیان الگوریتم بایستی سعی شود که دستورات ارائه شده بصورت عمومی و قانونمند باشد
- بطور مثال در الگوریتم خواندن کتاب دو نوع می توان بیان الگوریتم نمود:
 - خواندن صفحه اول سپس صفحه بعدی تا اینکه به انتها برسیم.
 - خواندن صفحه اول، صفحه دوم، صفحه سوم و

اجزای اصلی الگوریتم

- هر مساله راه حل و الگوریتم خاص خود را دارد.
- می توان برای حل یک مساله روشهای گوناگونی را ارائه داد.
- اما تمام الگوریتم ها دارای این اجزاء هستند:
 - نقطه شروع: حل مساله از کجا آغاز می گردد
 - فقط یک نقطه شروع در الگوریتم وجود دارد
 - نقطه پایان: جایی که مراحل حل مساله پایان می پذیرد.
 - به هر حال الگوریتم بایستی در یک نقطه خاتمه یابد.
 - می توان چندین نقطه پایان برای الگوریتم داشت.
- دستورالعملها و یا جملات اجرایی

متغیر (Variable)

- به خانه ای از حافظه که داده ها و اطلاعات ورودی یا خروجی و یا اطلاعات موقت را در خود نگه می دارد متغیر گفته می شود.
- مقدار متغیر می تواند در طول اجرای الگوریتم تغییر داشته باشد.

	1	1	1	0	0	0	0	1
i	1	1	0	1	0	0	0	1
N								

مثالی از یک الگوریتم

• الگوریتم محاسبه و چاپ مجموع دو عدد ۱۰ و ۲۰

. شروع

. عدد ۱۰ را در خانه (متغیر) A قرار بده

. عدد ۲۰ را در خانه B قرار بده

. محتویات خانه های A , B را با هم جمع کن و در خانه C قرار بده

. مقدار خانه C را بعنوان نتیجه چاپ کن

. پایان

استفاده از بیان ریاضی

- بیان الگوریتم در قالب جملات نوشتاری طولانی و فهم الگوریتم را دشوار می سازد.
- الگوریتم محاسبه و چاپ مجموع دو عدد ۱۰ و ۲۰
 - . شروع
 - . $A \leftarrow 10$
 - . $B \leftarrow 20$
 - . $C \leftarrow A+B$
 - . چاپ مقدار C
 - . پایان

انواع جملات مورد استفاده در الگوریتم ها

- جملات شرطی
- جملات عملیاتی (یا محاسباتی)
- جملات ورودی / خروجی
- جملات توضیحی

مثال

• الگوریتمی بنویسید که اعداد زوج دو رقمی را چاپ کند.

• (می دانیم که کوچکترین عدد زوج دو رقمی ۱۰ و اعداد زوج به اندازه ۲ واحد از هم فاصله دارند)

• شروع

• $J \leftarrow 10$

• J را چاپ کن

• $J \leftarrow J + 2$

• اگر $J \leq 98$ است آنگاه به مرحله ۳ برو

• پایان

مثال

• الگوریتمی بنویسید که اعداد زوج از ۱۰۰۰ تا ۲۰۰۰ را تولید و چاپ کرده و مجموع آنها را هم محاسبه کند.

. شروع

. $S \leftarrow 0$ ، $J \leftarrow 1000$

. J را چاپ کن و $S \leftarrow S + J$

. $J \leftarrow J + 2$

. اگر $J \leq 2000$ است آنگاه به مرحله ۳ برو در غیر اینصورت مقدار S را چاپ

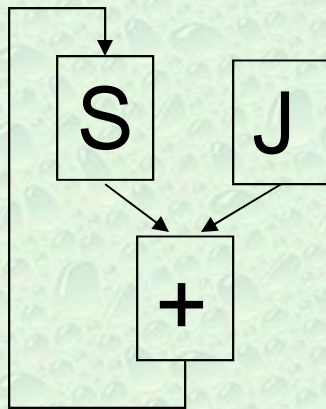
کن

. پایان

نکته

- اگر بخواهیم مقدار حاصل جمعی را محاسبه کنیم (مانند مثال قبل که مجموع اعداد زوج از ۱۰۰۰ تا ۲۰۰۰)

- ابتدا متغیری (مانند S) در نظر می گیریم و مقدار اولیه آن را صفر می گذاریم (یعنی اینکه هنوز هیچ مجموعی را حساب نکرده ایم) ، $S \leftarrow 0$
- سپس تک تک جملاتی را که قرار است با هم جمع کنیم را تولید کرده (با کمک یک متغیر دیگر، مثلا J در مثال قبلی) و با متغیر S جمع نموده و حاصل را در S قرار می دهیم (در واقع مقدار جدید را به حاصل قبلی می افزاییم - مانند انباره) ، $S \leftarrow S + J$
- در نهایت حاصل مجموع در این متغیر S قرار می گیرد.



ویژگیهای یک الگوریتم خوب

اگر چه یک مساله راه حل‌های مختلفی دارد، مهم یافتن بهترین راه حل است

- **سادگی**

- حتی الامکان ساده و عاری از ابهام و پیچیدگی باشد.

- **در نظر گرفتن تمام حالات خاص**

- الگوریتم بتواند در برابر حالات و شرایط مختلف پاسخ و جواب مناسبی ارائه دهد.

- بطور مثال هنگام حل معادله درجه دوم حالت‌های منفی زیر رادیکال را در نظر بگیرد.

- **روان بودن متن الگوریتم**

- دستورالعملها گویا بوده و منظور آنها بسادگی درک شود.

- **حداقل بودن تعداد دستورات و جملات**

ایجاد حلقه های تکرار (Loops)

- گاهی اوقات برای حل مساله باید یک یا چند مرحله از دستورات را تکرار نمود.
- به مرحله‌ای از الگوریتم که اجرای آنها چندین بار تکرار می‌شود حلقه (Loop) و یا حلقه تکرار گفته می‌شود
- بطور کلی حلقه های تکرار از اجزای زیر تشکیل شده است:

– شمارنده حلقه (Counter)

- یک متغیر کمکی که پیش از شروع حلقه به آن مقدار اولیه داده می‌شود
- از طریق آن می‌توان تعداد دفعات تکرار حلقه را نشان داد.

– گام افزایش (Step)

- مقداری که پس از هر بار مراحل حلقه به شمارنده اضافه می‌شود.

– شرط پایانی

- مقدار و یا متغیری است که پس از اجرای دستورات حلقه با شمارنده حلقه مقایسه می‌گردد و زمان پایان اجرای دستورات حلقه را مشخص می‌سازد.

– بدنه حلقه

- دستورالعملها و جملاتی که عملیات اصلی حلقه را تشکیل می‌دهند.

استفاده از الگوریتم های فرعی

- با بزرگتر و پیچیده تر شدن مساله، الگوریتم آن نیز پیچیده و بزرگتر خواهد شد.
- پیچیده شدن الگوریتم چنین مشکل بوجود می آورد:
 - از وضوح و روانی الگوریتم می کاهد
 - منطق الگوریتم را دشوارتر می سازد.
 - اشکال زدایی (Debug) و پیدا کردن نقاط ضعف الگوریتم را سخت تر می سازد.
- برای رفع این مشکلات از الگوریتم های فرعی استفاده می شود
 - به این معنی که ابتدا مساله را به تعدادی بخش های مختلف و مستقل تقسیم نمود
 - و برای هر کدام الگوریتم جداگانه ای نوشت

استفاده از الگوریتم های فرعی

مثال : الگوریتمی بنویسید که با دریافت دو عدد طبیعی m, n مقدار C_n^m را محاسبه کند.

$$C_n^m = \frac{m!}{n!(m-n)!}$$

بهترین راه حل این مساله این است که:

الگوریتم فرعی برای محاسبه فاکتوریل نوشتن مثلاً با نام **Fact(z)** که با دریافت z بعنوان ورودی، فاکتوریل آنرا محاسبه کرده و بر می گرداند.

و در برنامه اصلی از این الگوریتم اصلی استفاده نمود.

$$A = \text{Fact}(m), \quad B = \text{Fact}(n), \quad C = \text{Fact}(m-n)$$

$$C_n^m = A / (B \times C)$$

فلوچارت (Flowchart) یا نمودار گردش

- فلوچارت، بیان تصویری الگوریتم با کمک مجموعه ای استاندارد از اشکال ساده می باشد
- فلوچارت یکی از روشهای برقراری ارتباط منطقی بین مراحل مختلف حل مساله است.
- اشکال استاندارد موجود:

