



# دانشگاه می علوم ریاضی و آمار



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

مبانی کامپیوتر و برنامه سازی

جلسه ۱۷

نگارنده: علیرضا چگنی

۱۹ آبان ۱۴۰۰

مسائل محاسباتی ..... ۲

الگوریتم ..... ۲

## ۱ مسائل محاسباتی

کار اصلی یک CS حل مسائل محاسباتی است. مسائل محاسباتی را از حیث قابلیت حل پذیری به دو رده زیر تقسیم می کنند:

- غیر قابل حل : برای آن هیچ الگوریتمی نمی تواند وجود داشته باشد. مانند مسئله توقف (halting problem).

- قابل حل : مسائلی که برای آنها الگوریتمی وجود دارد. مسائل قابل حل نیز به دو رده کلی قابل تقسیم هستند:

- قابل حل در تئوری: در عمل نمی توان آنها را برای نمونه های بزرگ به کار گرفت، مانند دور همیلتونی.

- قابل حل در عمل: در عمل برنامه های این کلاس قابل استفاده هستند، مانند مرتب کردن یک لیست از اعداد.

## ۲ الگوریتم

یک الگوریتم مجموعه ای از اعمال گام به گام است که هدف معینی را دنبال می کند و به طور کلی ویژگی های زیر است :

(۱) ورودی (Input)

(۲) خروجی (Output)

(۳) عمومیت (Generallity): قابل اعمال روی مجموعه ای از ورودی هاست،

(۴) متناهی (Finiteness) : بعد از تعداد گام متناهی متوقف می شود،

(۵) دقت (Precision) : گام ها به طور دقیق تعریف شده اند،

(۶) صحت یا درستی (Correctness): خروجی تولید شده توسط الگوریتم درست است.

مثال ۱- انجام محاسبه جمع دو عدد،

۲- انجام محاسبه تفریق دو عدد دلخواه،

۳- انجام محاسبه یک عملگر دو عملوندی دلخواه،

الگوریتم یک چارچوب یا قالب برای بیان مسئله حل شده است و نحوه نمایش آن می تواند به یکی از صورت های زیر باشد:

۱- بیان در قالب متن فارسی یا انگلیسی

۲- بیان در قالب فلوچارت

۳- بیان در قالب شبه کد

۴- بیان در قالب یک برنامه در یک زبان برنامه نویسی

نکته: مورد ۱ و ۲ بیشتر ماهیت آموزشی دارد و بیان رسمی الگوریتم ها بیشتر در قالب شبه کد می باشد.

### مثال

در قالب متن فارسی: محاسبه جمع دو عدد دلخواه

گام ۱ : شروع،

گام ۲ : عدد دلخواه اول را وارد کن،

گام ۳ : عدد دلخواه دوم را وارد کن،

گام ۴ : مجموع دو عدد دریافت شده را حساب کن،

گام ۵ : حاصل جمع را اعلام کن،

گام ۶ : پایان.

Addition of two arbitrary numbers

Step1: Start,

Step 2: Take the first number,

Step 3: Take the second number,

Step 4: Compute the sum of two received numbers,

Step 5: Announce the result of Step4,

Step 6: End.

به طور کلی سه ساختار کنترلی برای نوشتن برنامه ، شبه کد و فلوچارت وجود دارد:

۱- دنباله (Sequence)

۲- انشعاب (Branching) که انتخاب (Selection) نیز گفته می شود.

۳- حلقه تکرار (Repetition)

اکنون سه ساختار فوق را به تفصیل مورد بحث قرار می دهیم:

❖ دنباله (Sequence): توالی از دستورات که یکی پس از دیگری می آیند:

➤ خواندن در ورودی : Read

➤ نوشتن در خروجی : Print

➤ محاسبات: محاسبات ریاضی منطقی و ...

مثال

خواندن یک عدد و چاپ دو برابر آن در خروجی :

- 1- Start,
- 2- Read A,
- 3-  $C = A * 2$ ,
- 4- print C ,
- 5- End.

❖ انشعاب (Branching): یک تقسیم دودویی بر پایه یک شرط (condition) است. از عبارت زیر در گام

های الگوریتم استفاده می کنیم:

دستور    then شرط    If

مثال

دریافت دو عدد از ورودی و چاپ بزرگترین آنها در خروجی

- 1- Start,
- 2- Read A,B,
- 3- If A greater than B,
- 4- If B greater than A then  $C=B$ ,
- 5- Print C,
- 6- End.

❖ حلقه (Loop): اجازه می دهد یک دستور یا توالی مکرر از دستورات بر پایه یک شرط اجرا به طور مکرر

شوند. برای ایجاد حلقه از قالب زیر استفاده می کنیم:

If شرط then go to line #n

جائیکه #n بیانگر یک شماره خط قبل از خط جاری است.

- 1- Start,
- 2-  $I = 0$ ,
- 3- Print I,
- 4-  $I = I + 2$ ,
- 5- If ( $I \leq 98$ ) then go to line 3,
- 6- End.