



درس «مبانی کامپیوتر و برنامه‌سازی»

الگوریتم و فلوچارت

صادق علی اکبری

سرفصل مطالب

● حل مسأله

● الگوریتم

● شبه کد

● فلوچارت



حل مسئله (Problem Solving)

- حل مسئله نیازمند:
 - توان تجزیه و تحلیل صورت مسئله
 - دانش و تجربه حل مسئله در حوزه موردنظر
 - از اختراع چرخ بهره‌زید!
 - خلاقیت
- مهندس خوب: کسی که توان حل مسئله بالایی در یک حوزه فناوری دارد

حل مسئله در برنامه‌نویسی

- برنامه‌نویسی: یک روش حل مسئله (Problem Solving)
- مسئله: نرم‌افزار موردنیاز
 - برنامه‌ی یک بازی، یک وبسایت، یک نرم‌افزار هوشمند و یا ...
- هرچند دانش «برنامه‌نویسی» برای حل مسئله بسیار مهم است، اما برنامه‌نویسی یکی از آخرین گام‌ها در این راه است
- طراحی راه‌حل، قبل از پیاده‌سازی و آزمایش راه‌حل انجام می‌شود

- ۱- تحلیل و درک دقیق صورت مساله
- فهم مفروضات مسأله و ارتباط منطقی بین معلومات و مجهولات
- ۲- طراحی مناسب‌ترین راه‌حل (الگوریتم)
- با کمک ابزارهای انتزاعی‌تر (سطح بالاتر) مانند شبه‌کد یا فلوچارت
- ۳- بررسی و ارزیابی راه‌حل
- ۴- نوشتن برنامه کامپیوتری
- ۵- آزمایش برنامه کامپیوتری

تعریف الگوریتم (Algorithm)

- دستورالعملی که مراحل انجام کاری را با زبانی دقیق و با جزئیات کافی بیان نماید به طوری که ترتیب مراحل و شرط خاتمه عملیات در آن کاملاً مشخص باشد
- مجموعه‌ای متناهی از دستورالعمل‌ها که به ترتیب خاصی اجرا می‌شوند و مسأله‌ای را حل می‌کنند



- واژه‌ی الگوریتم (الخوارزمی) از نام ریاضیدان و فیلسوف ایرانی، ابوجعفر محمد بن موسی خوارزمی (قرن سوم هجری)، گرفته شده است

- الگوریتمی بنویسید که دو عدد از ورودی دریافت کرده مجموع دو عدد را محاسبه و چاپ نماید (نمایش دهد)

<u>خروجی‌ها</u>	<u>انجام محاسبات</u>	<u>ورودی‌ها</u>
مجموع دو عدد	جمع دو عدد	a, b

- ۰- شروع
- ۱- a و b را بخوان.
- ۲- مجموع a و b را محاسبه و در sum قرار بده.
- ۳- sum را در خروجی چاپ کن
- ۴- پایان

• در طراحی الگوریتم‌های ساده، معمولاً سه فعالیت اصلی وجود دارد:

• خواندن داده‌ها

• انجام محاسبات

• نمایش خروجی‌ها

• استفاده از متغیرها:

• برای دریافت داده‌ها (متغیرهای ورودی)

• برای محاسبات (متغیرهای میانی)

• برای خروجی‌ها (متغیرهای خروجی)

روش‌های توصیف الگوریتم

- متن و نوشتار

- بدون ساختار مشخصی، به صورت متنی و محاوره‌ای الگوریتم را توصیف کنیم
- (معمول و رایج نیست)

- نمودار

- مثلاً فلوچارت

- با سمبل‌های گرافیکی روند الگوریتم را تعیین کنیم

- شبه‌کد (Pseudocode)

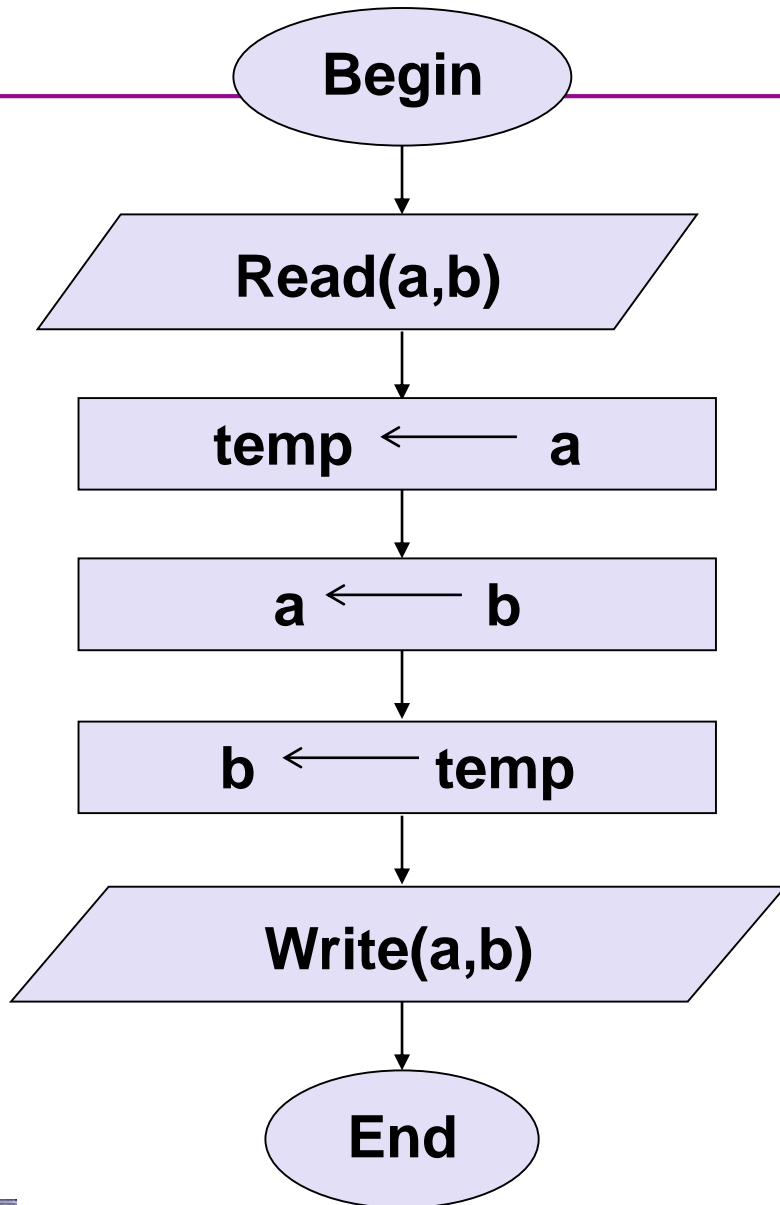
- مشابه یک برنامه کامپیوتری، ولی از گرامر زبان خاصی پیروی نمی‌کند

شبه‌کد (Pseudocode)

```
IF (a > 10) AND (b = 5)
  THEN PRINT "Hello there!"
ELSE IF (c = d) OR (a = d)
  THEN PRINT "Goodbye"
  ELSE PRINT "My head hurts"
ENDIF
ENDIF
```

- برنامه‌ای با یک زبان ساختگی و غیررسمی که به برنامه‌نویس در توسعه‌ی الگوریتم کمک می‌کند
- یک زبان برنامه‌نویسی واقعی نیست ولی کار با آن ساده است
- هنگام نوشتن شبه‌کد، درگیر جزئیات گرامر یک زبان برنامه‌نویسی خاص نمی‌شویم
- شبه‌کد خوب را به سادگی می‌توان با یک زبان برنامه‌نویسی دلخواه پیاده‌سازی نمود
- جزئیات لازم باید در شبه‌کد ذکر شده باشد
- ساختار کلی شبه‌کد مشابه یک برنامه کامپیوتری است

فلوچارت (Flowchart)



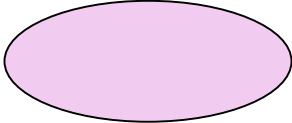



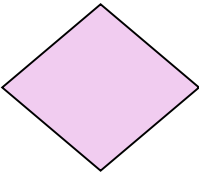
- روندنما یا فلوچارت

- نموداری که روند اجرای برنامه را مشخص می کند

- مثال:

- فلوچارت روبرو چه الگوریتمی را نشان می دهد؟

مؤلفه‌های نمودار فلوچارت

مثال	شرح	شکل
<p>start</p> <p>stop</p>	شروع و خاتمه الگوریتم	
<pre>c ← a + b d ← i</pre>	محاسبات و عملیات انتساب	
<p>A, B</p>	ورود و خروج اطلاعات برای کاربر (صفحه نمایش)	
<p>A, B, "100"</p>	پرینت اطلاعات بر روی کاغذ	
<p>ورودی</p> <p>خروجی</p> <p>خروجی</p>	تصمیم‌گیری و شرط	

مثال: الگوریتمی که دو عدد از ورودی بگیرد و حاصل ضرب آن دو را محاسبه کند و نمایش دهد

چند نمونه شبه کد

- مقدار a و b را بخوان
- حاصل ضرب a و b را در c قرار بده
- مقدار c را چاپ کن

Start
Read X, Y
 $M \leftarrow X * Y$
PRINT M
END

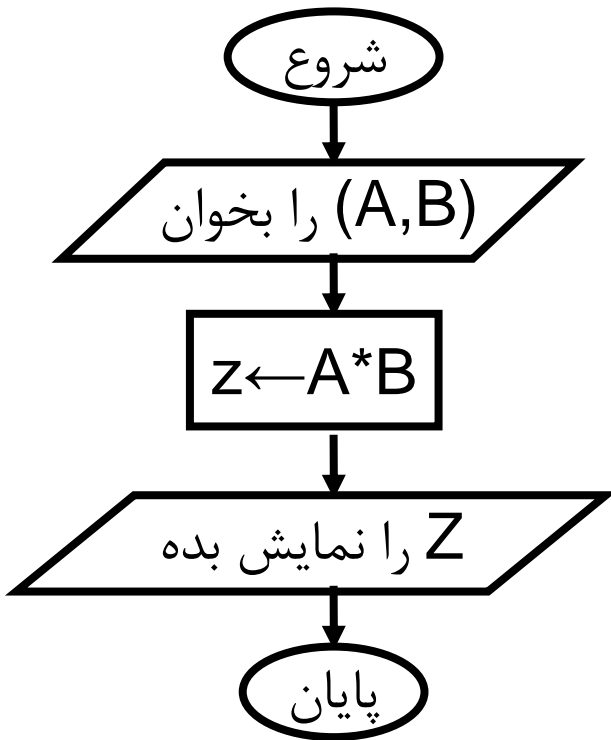
Begin
Input a, b
 $c := a * b$
Output c
Finish

کد (برنامه C)

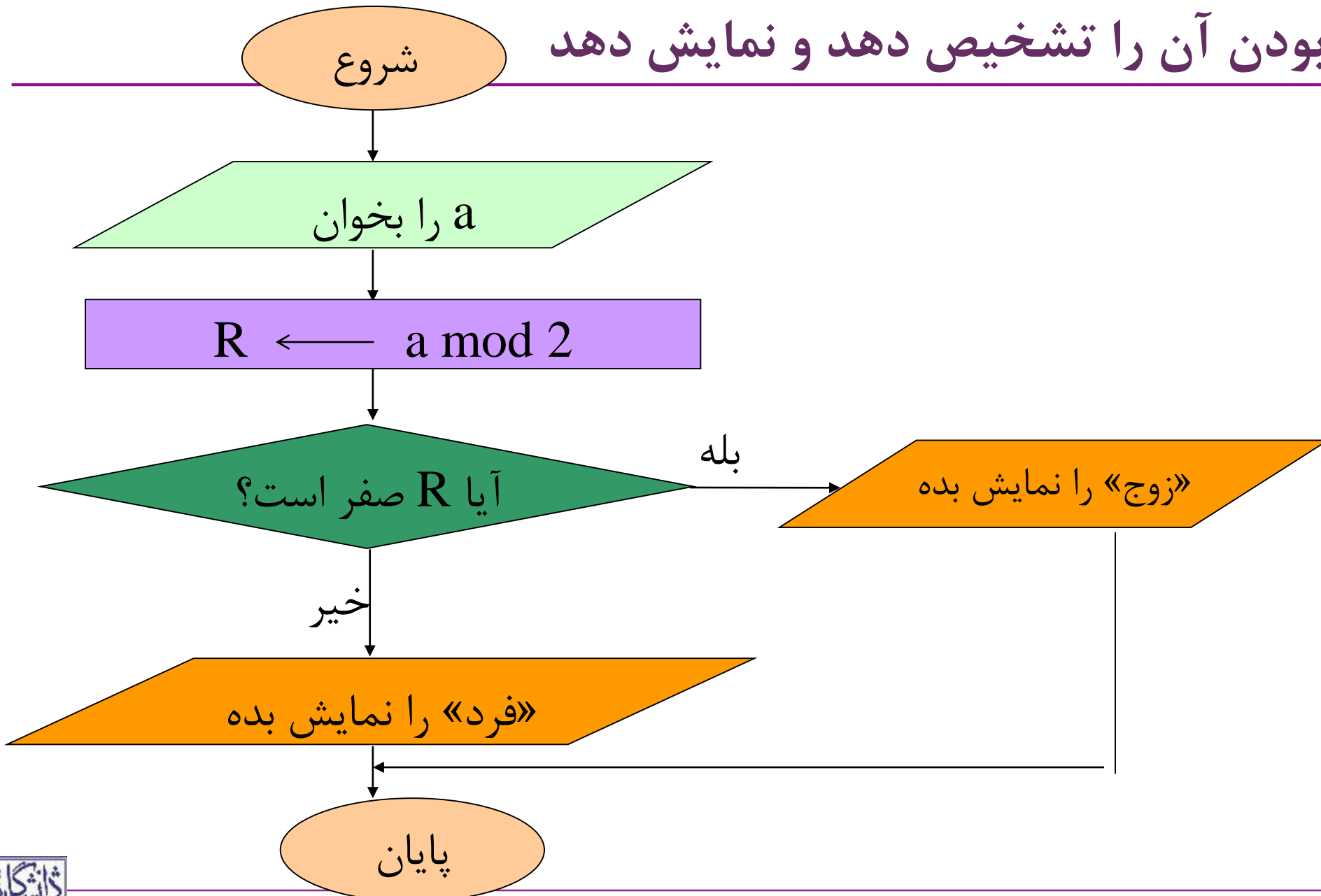
```
#include <stdio.h>
int main() {
    int a , b;
    scanf("%d", &a);
    scanf("%d", &b);
    int c = a*b;
    printf("%d", c);

    return 0;
}
```

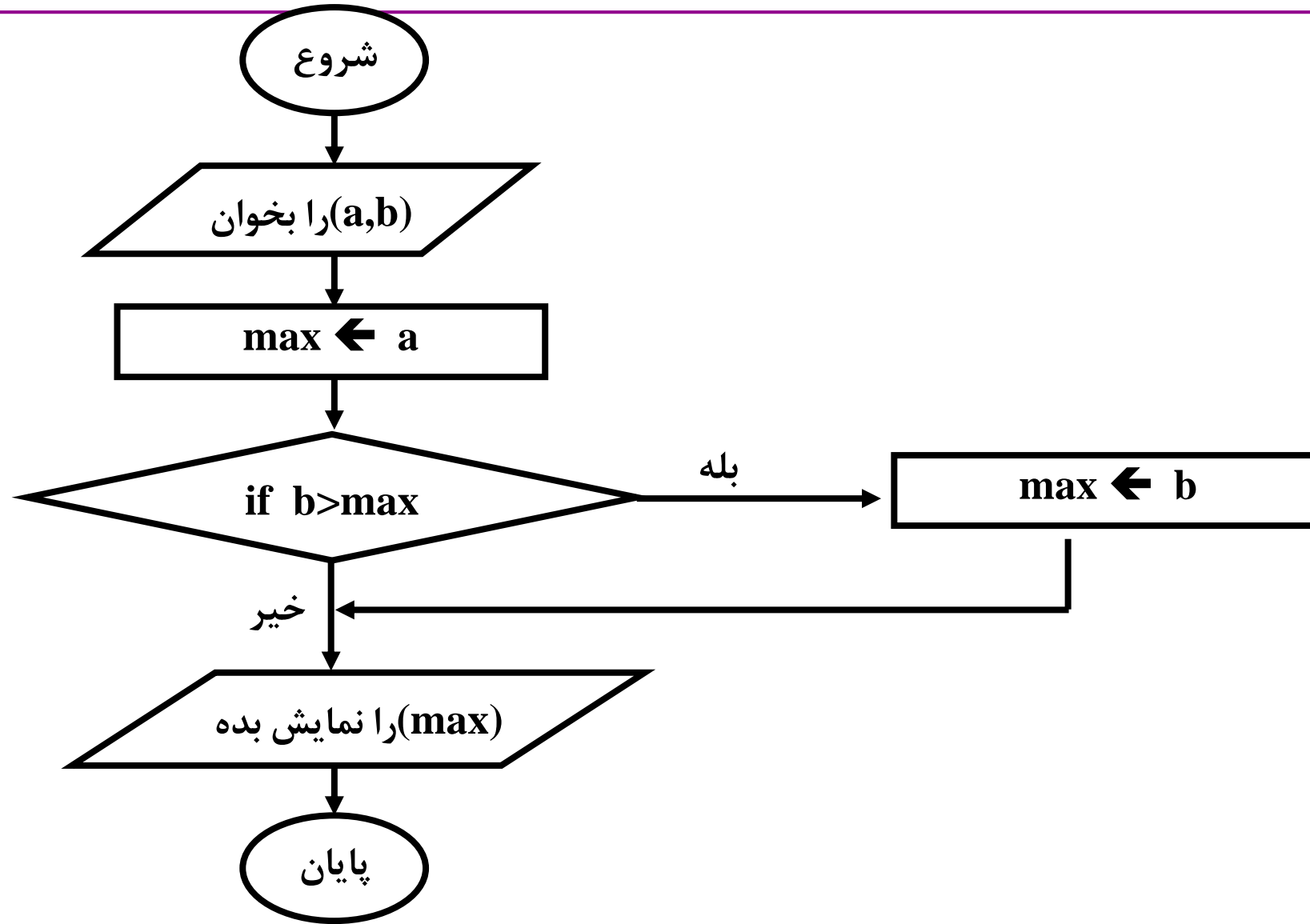
فلوچارت



فلوچارتی که عددی را از ورودی دریافت کند و فرد یا زوج بودن آن را تشخیص دهد و نمایش دهد

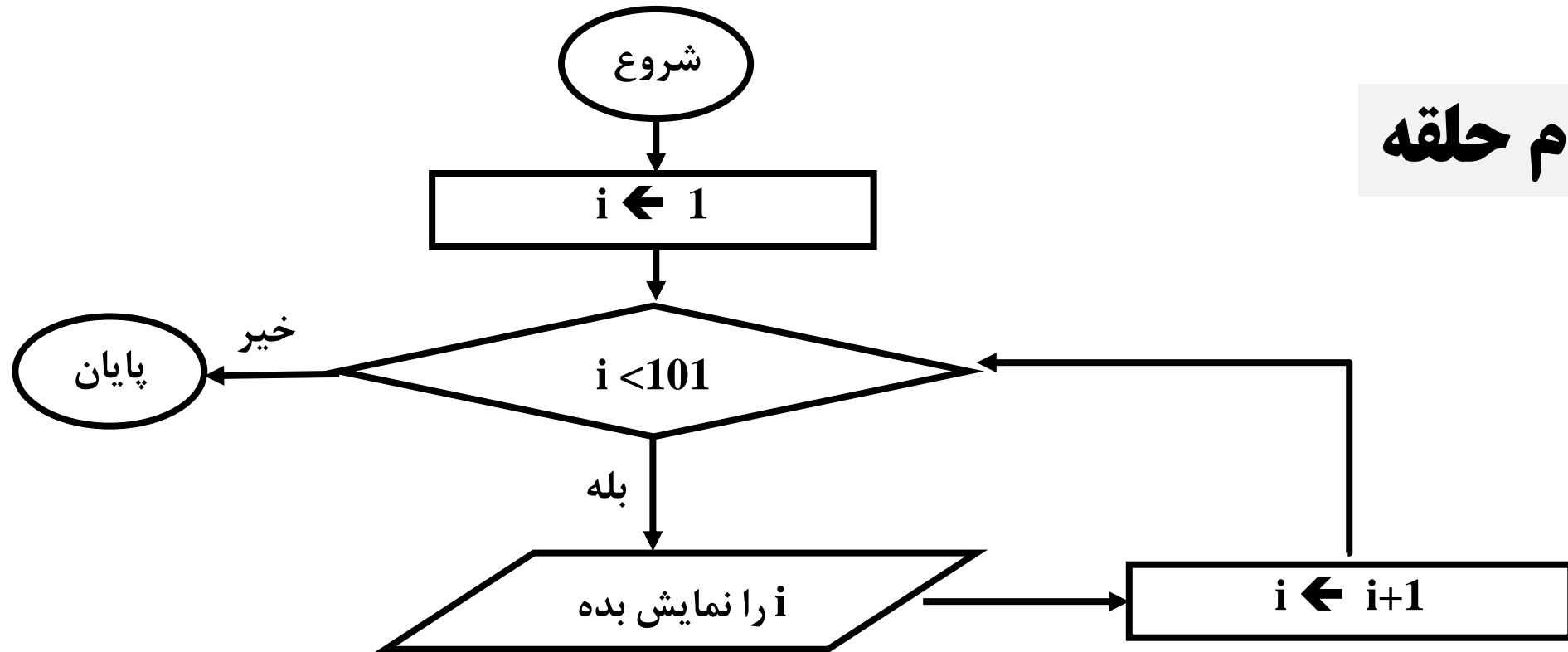


مثال : فلوچارتی که دو عدد از ورودی دریافت کند، بزرگترین عدد را پیدا کند و در خروجی نمایش دهد

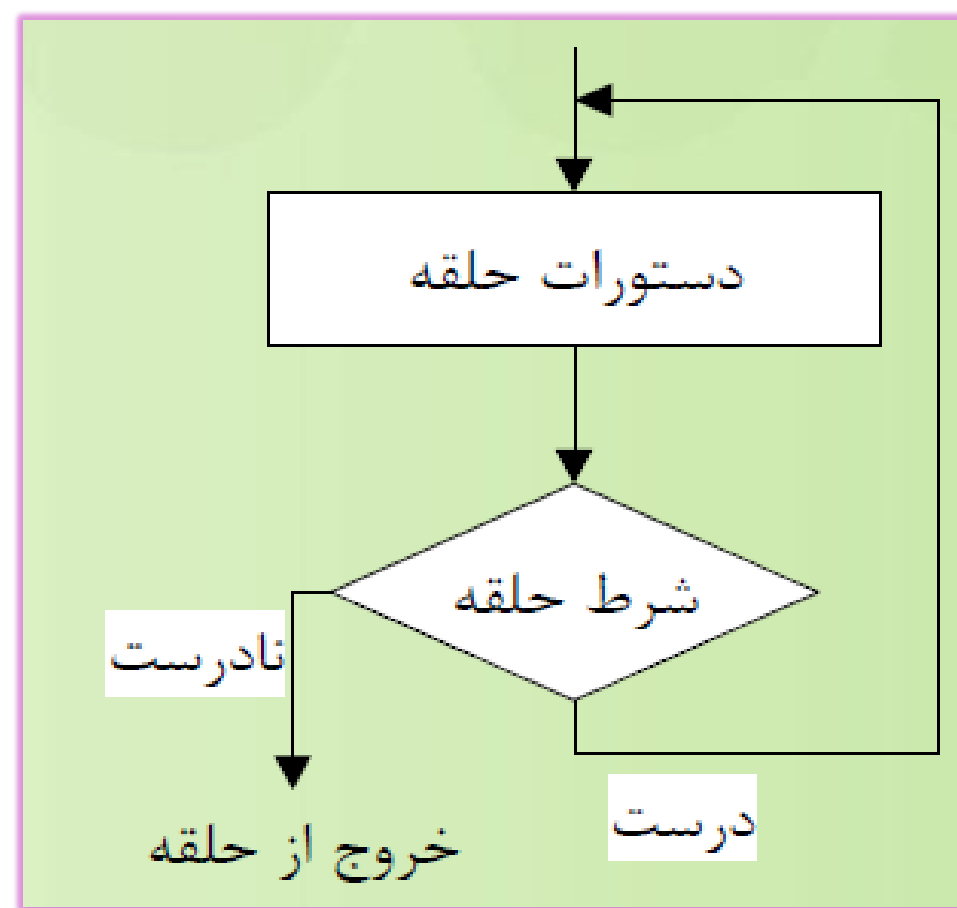
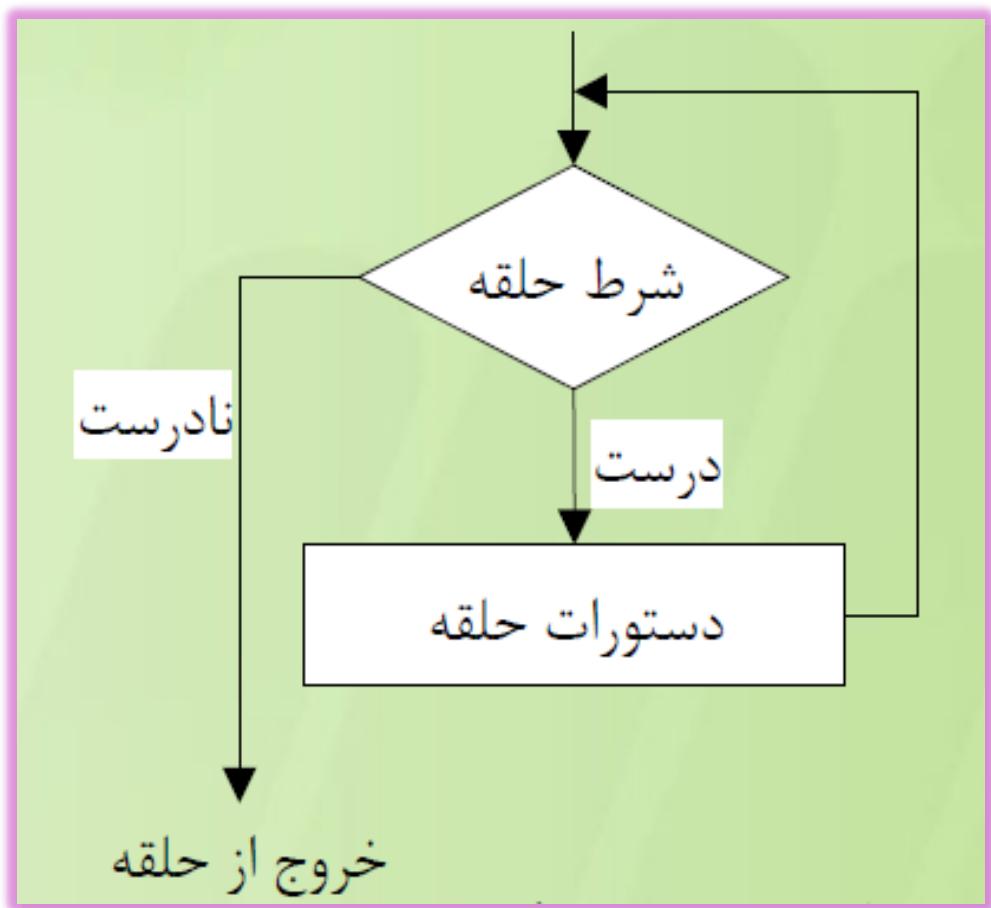


فلوچارتی که اعداد یک تا ۱۰۰ را چاپ کند

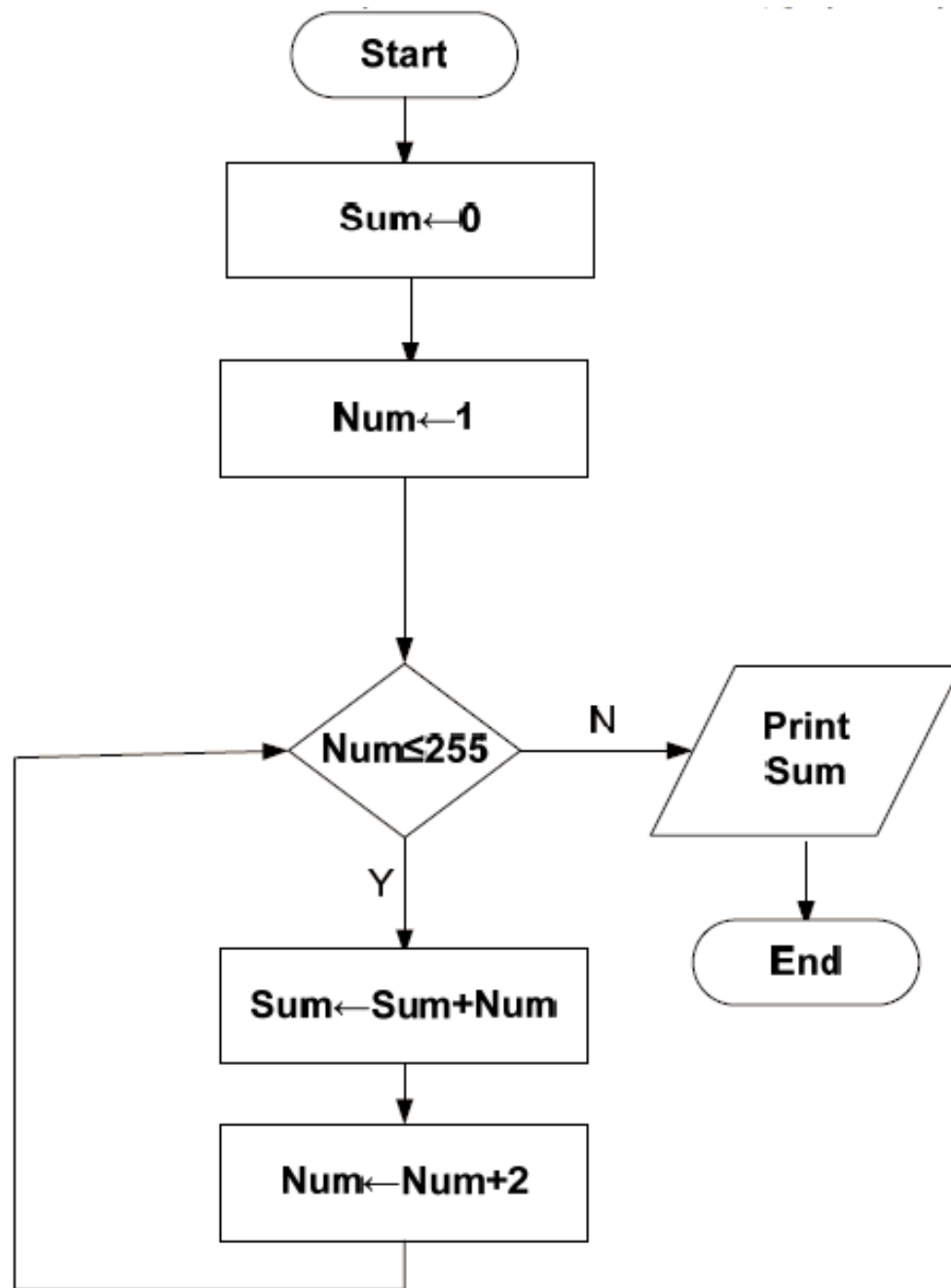
مفهوم حلقه



نحوه ایجاد حلقه (Loop)



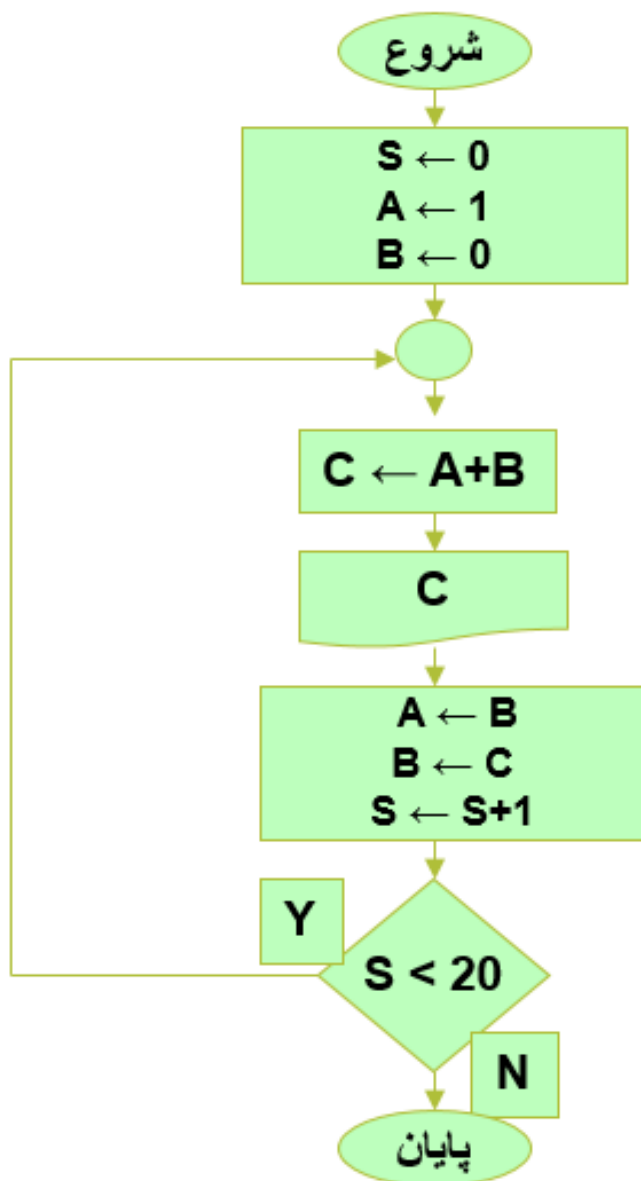
فلوچارتی که
مجموع اعداد فرد
از یک تا ۲۵۵ را حساب کند
(بدون استفاده از فرمول تصاعد)



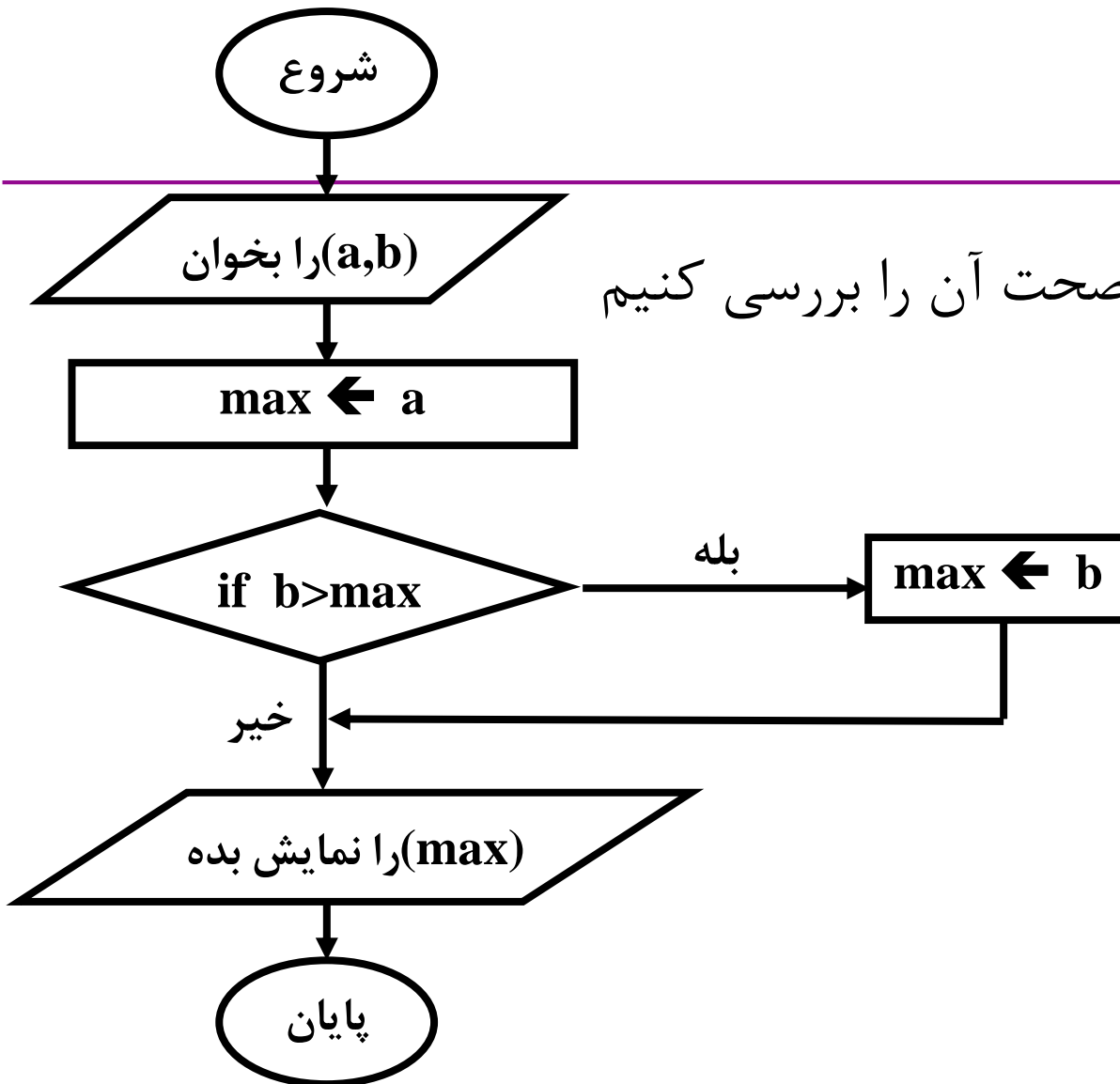
دستورات مجاز در نوشتن الگوریتم

- ورودی و خروجی (بخوان و بنویس)
- محاسبات و عملگرها (مثلاً ضرب و تقسیم)
- انتساب (تعیین مقدار یک متغیر)
- شرط (اگر ... آن گاه ... وگرنه ...)
- حلقه
- دستور «برو به» (goto)
- مثال: اگر $a > 2$ برو به مرحله سوم از الگوریتم
- فقط برای ایجاد شرط و حلقه از این دستور استفاده کنید (در برنامه‌ها هم اصلاً استفاده نکنید!)

- الگوریتمی که عدد N را از کاربر بگیرد و سپس N بار اعدادی را از ورودی خوانده مجموع آنها را محاسبه کرده و چاپ نماید.
- الگوریتمی که یک عدد صحیح مثبت را از ورودی خوانده تعداد ارقام آن را در خروجی چاپ کند
- الگوریتمی که سری فیبوناچی را به ازای مقادیر کوچکتر از ۱۰۰ تولید و چاپ نماید: ۱, ۱, ۲, ۳, ۵, ۸, ۱۳, ۲۱, ...
- الگوریتمی که میانگین جملات کوچکتر از ۱۰۰ سری فیبوناچی را محاسبه و چاپ نماید



تحلیل الگوریتم



• یعنی الگوریتم را دنبال (trace) کنیم و صحت آن را بررسی کنیم

• مثال: فرض کنید کاربر ۵ و ۷ را وارد کند

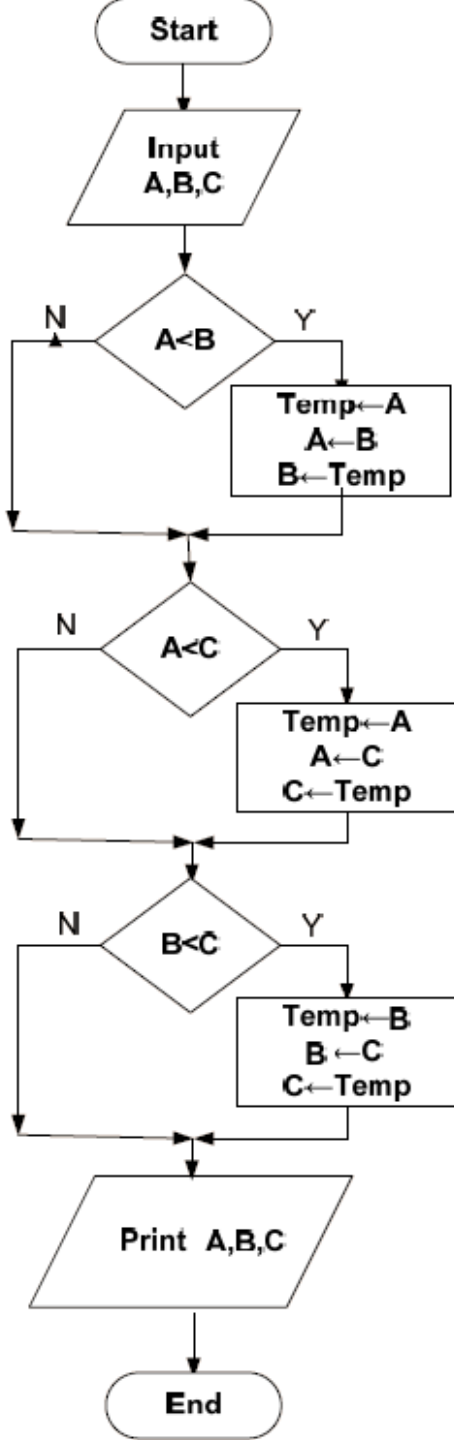
دستور	max	b	a	خروجی
a, b را بخوان	?	7	5	
$\text{max} \leftarrow a$	5	7	5	
$b > \text{max}$	5	7	5	
$\text{max} \leftarrow b$	7	7	5	
max را نمایش بده	7	7	5	7

• فلوجارتی که سه عدد از ورودی بگیرد
و آن‌ها را به ترتیب نزولی چاپ کند

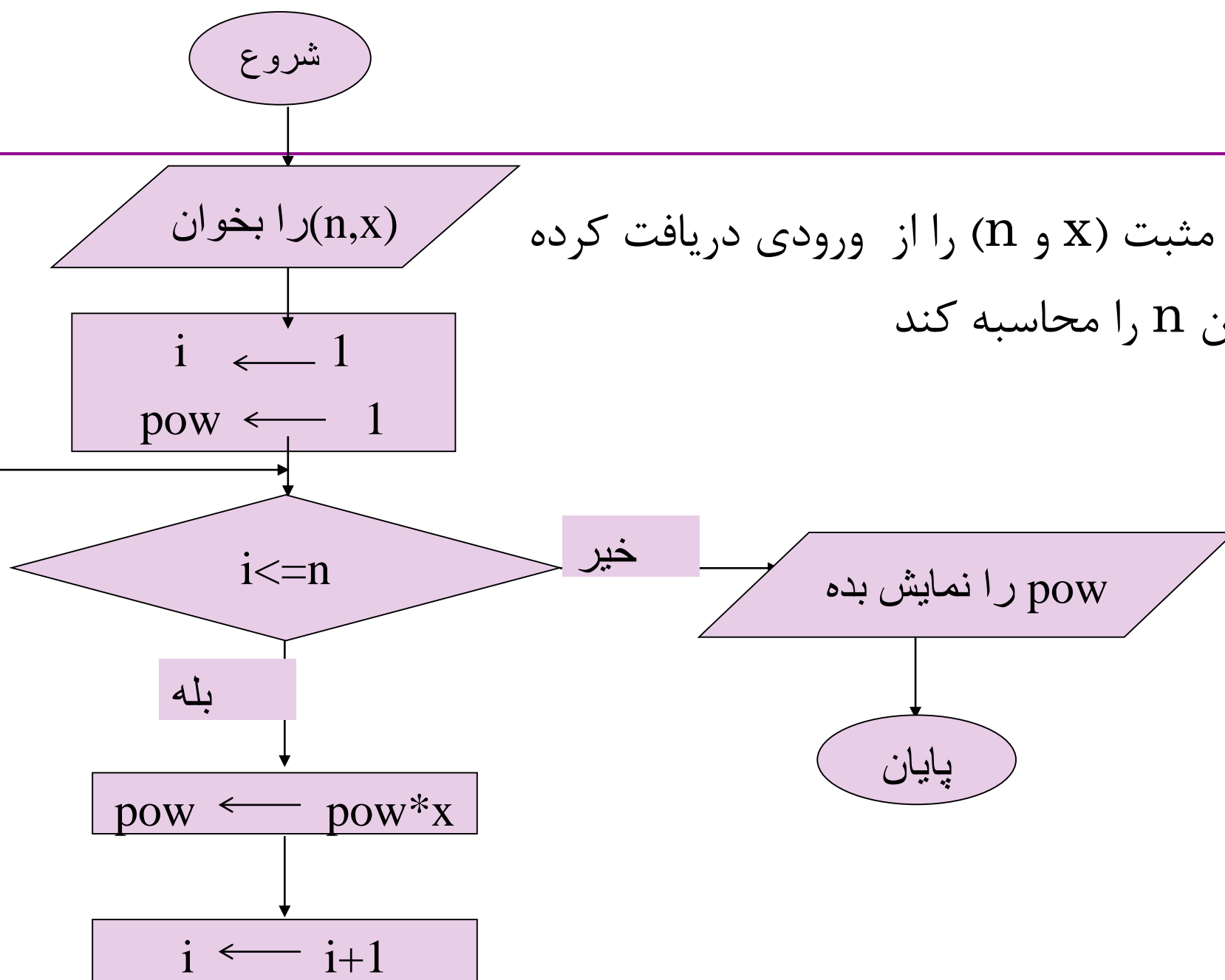
• الگوریتم را به ازای ورودی‌های زیر دنبال کنید

• ۱ و ۲ و ۳

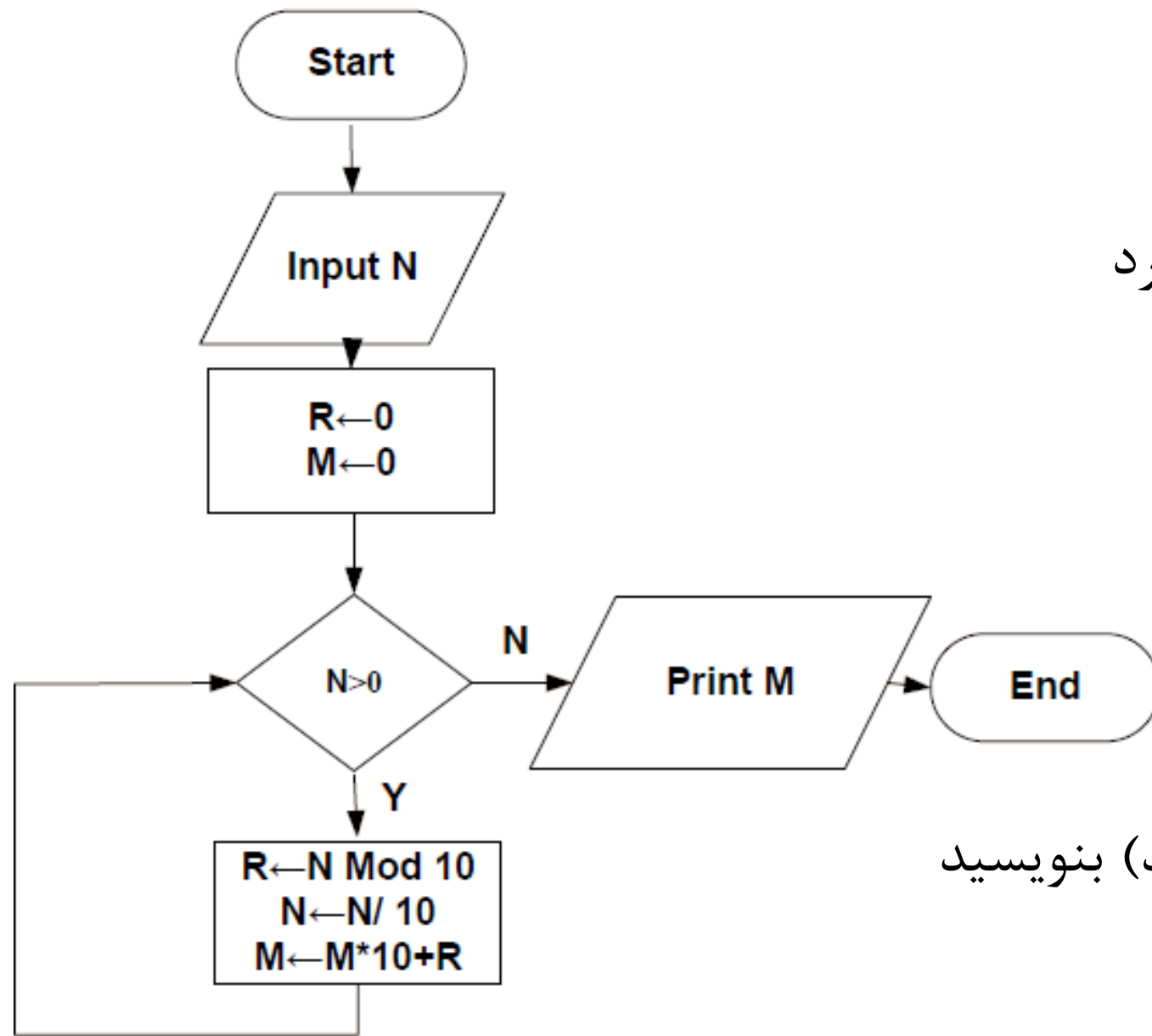
• ۳ و ۲ و ۱



- دو عدد صحیح مثبت (n و x) را از ورودی دریافت کرده سپس x به توان n را محاسبه کند



- فلوچارتی که یک عدد طبیعی بگیرد و مقلوب آن را چاپ کند
- مثلاً مقلوب ۱۳۶۷ می‌شود ۷۶۳۱
- این فلوچارت را تحلیل کنید
- مثلاً به ازای ۱۳۶۷
- به جای فلوچارت الگوریتم (شبه‌کد) بنویسید



- مثال دیگر: فلوچارتی که یک عدد بگیرد و بگوید آیا این عدد با مقلوبش برابر است یا خیر

- اسلایدهای بسیار خوب دکتر محمودی
- http://faculties.sbu.ac.ir/~a_mahmoudi/ITP_93_1.htm
- اگر مرور اسلایدهای درس و اسلایدهای دکتر محمودی کافی نبود و خوب یاد نگرفتید:
- کتاب «الگوریتم و فلوچارت»، نوشته: مهندس ایرج صادقی، انتشارات ناقوس

پایان