



مبانی برنامه نویسی

الگوریتم و فلوچارت

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir



<https://www.aparat.com/mehran.safayani>



https://github.com/safayani/Programming_Basics_course



۱-۳ تعریف الگوریتم

الگوریتم دستورالعملی برای حل مسئله است که دارای شرایط زیر باشد:

۱. به زبان دقیق گفته شود.
۲. جزئیات کامل حل مسئله را داشته باشد.
۳. ترتیب مراحل آن مشخص باشد.
۴. شرط خاتمه عملیات مشخص باشد.

۳-۵ بیان الگوریتم با جملات فارسی (یا هر زبان طبیعی دیگر)

در این روش، الگوریتم‌ها بدون استفاده از نمادهای خاصی ذکر می‌شوند. نمونه‌هایی از این الگوریتم‌ها را در ادامه مشاهده می‌کنید.

◀ مثال ۱-۳

الگوریتمی که مجموع دو عدد را محاسبه می‌کند.

۱. اولین عدد را انتخاب کرده، بر روی کاغذ بنویسید.
۲. دومین عدد را گرفته، آن را زیر عددی بنویسید که بر روی کاغذ نوشته شود.
۳. دو عدد روی کاغذ را با هم جمع کنید، زیر آن دو عدد بنویسید.
۴. سومین عددی که بر روی کاغذ نوشته شد، مجموع دو عدد است.
۵. پایان.

٣-٦ بیان ریاضی الگوریتم

◀ مثال ۳-۳

الگوریتمی که سه مقدار عددی را از ورودی خوانده، میانگین آنها را محاسبه می‌کند. در این الگوریتم باید برای سه مقداری که از ورودی خوانده می‌شوند، نامی انتخاب کنیم. نام‌گذاری برای میانگین و مجموع اعداد نیز لازم است.

متغیرهای الگوریتم

A	عدد اول
B	عدد دوم
C	عدد سوم
SUM	مجموع اعداد
AVE	میانگین اعداد

۱. A، B، C را از ورودی بخوان

۲. $SUM \leftarrow A + B + C$

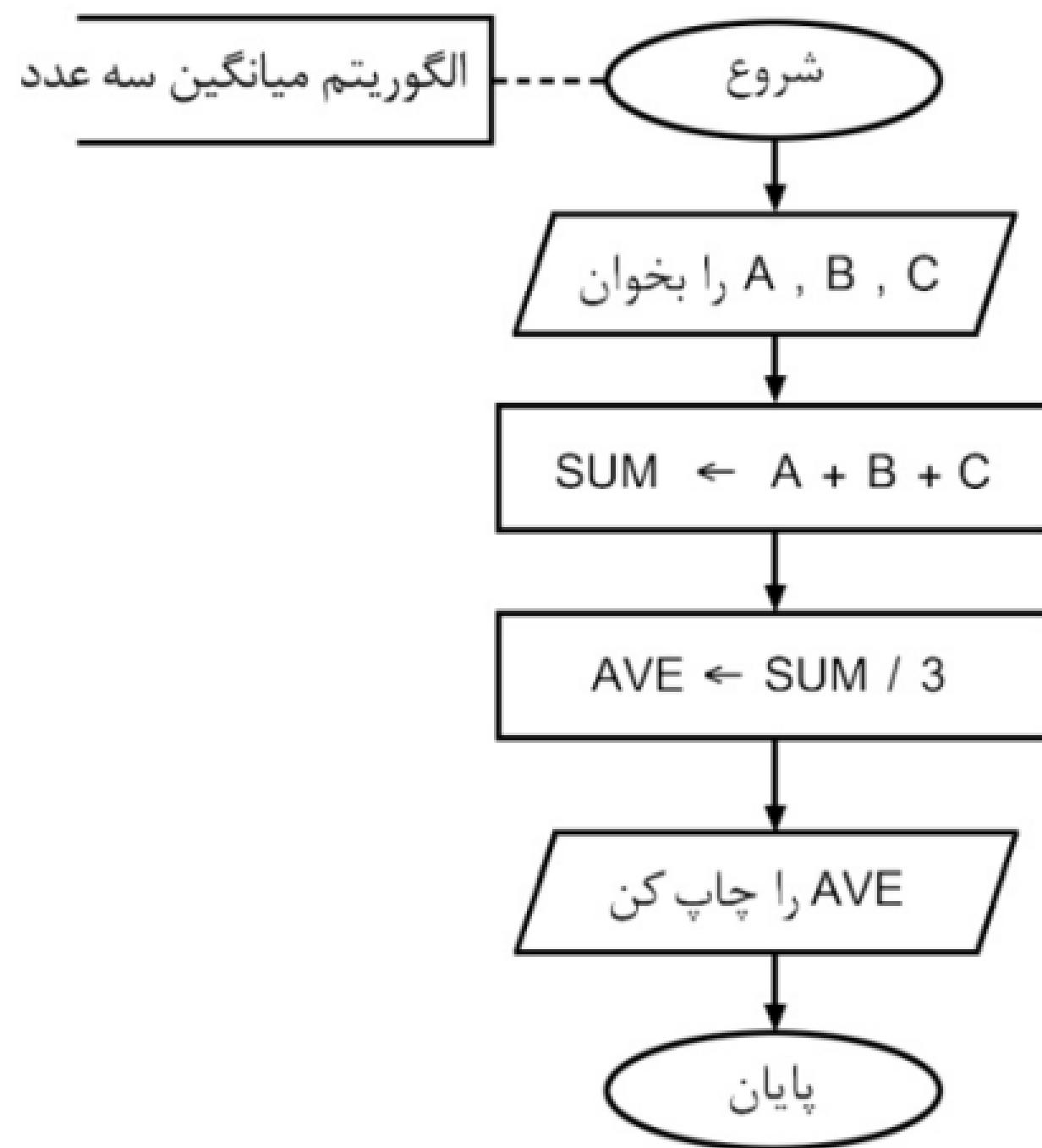
۳. $AVE \leftarrow SUM / 3$

۴. AVE را چاپ کن

۵. پایان

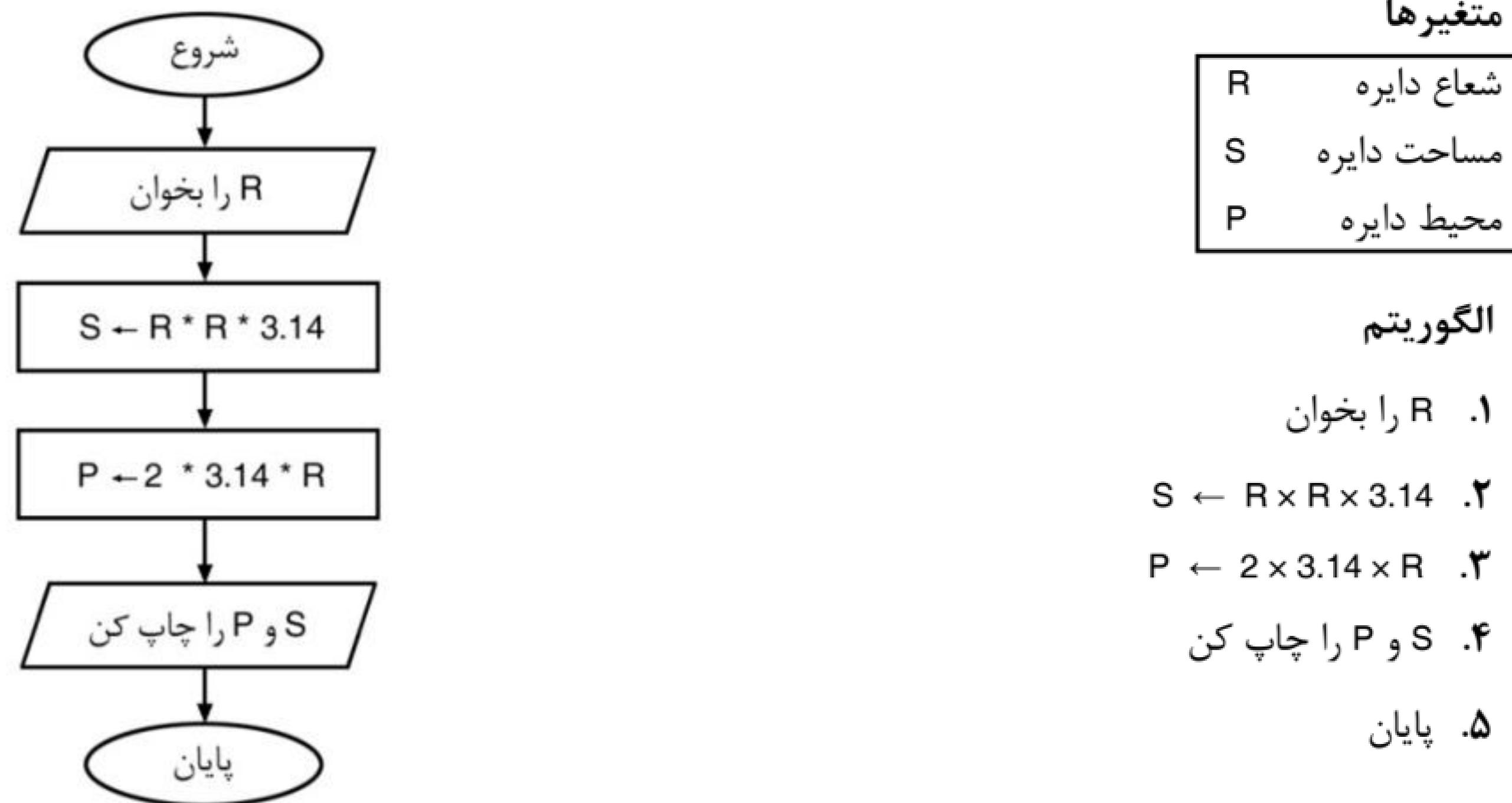
۷-۳ بیان الگوریتم توسط شکل‌ها

۸-۳ فلوچارت

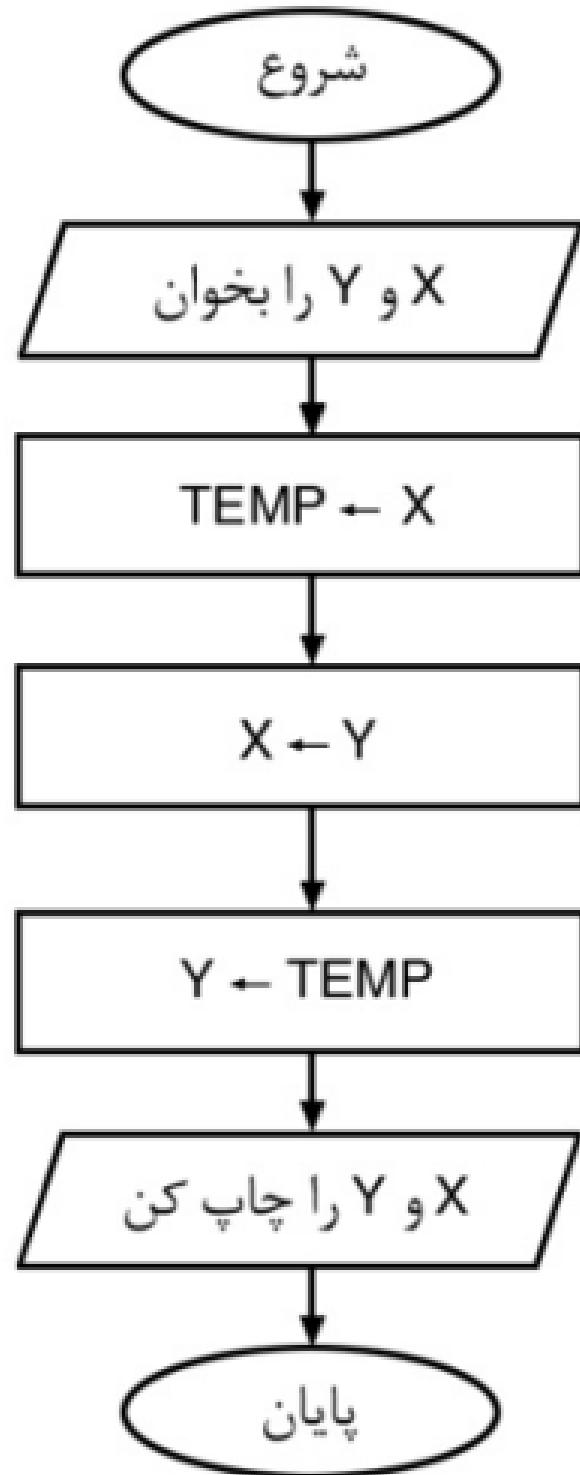


◀ مثال ۳-۴

الگوریتمی که شعاع دایره‌ای را از ورودی خوانده، محیط و مساحت آن را محاسبه می‌کند و به خروجی می‌برد.



◀ مثال ۳-۸



الگوریتمی که دو مقدار را از ورودی خوانده، در دو متغیر X و Y قرار می‌دهد و سپس محتويات آن دو را با هم عوض کرده، در خروجی چاپ می‌کند.

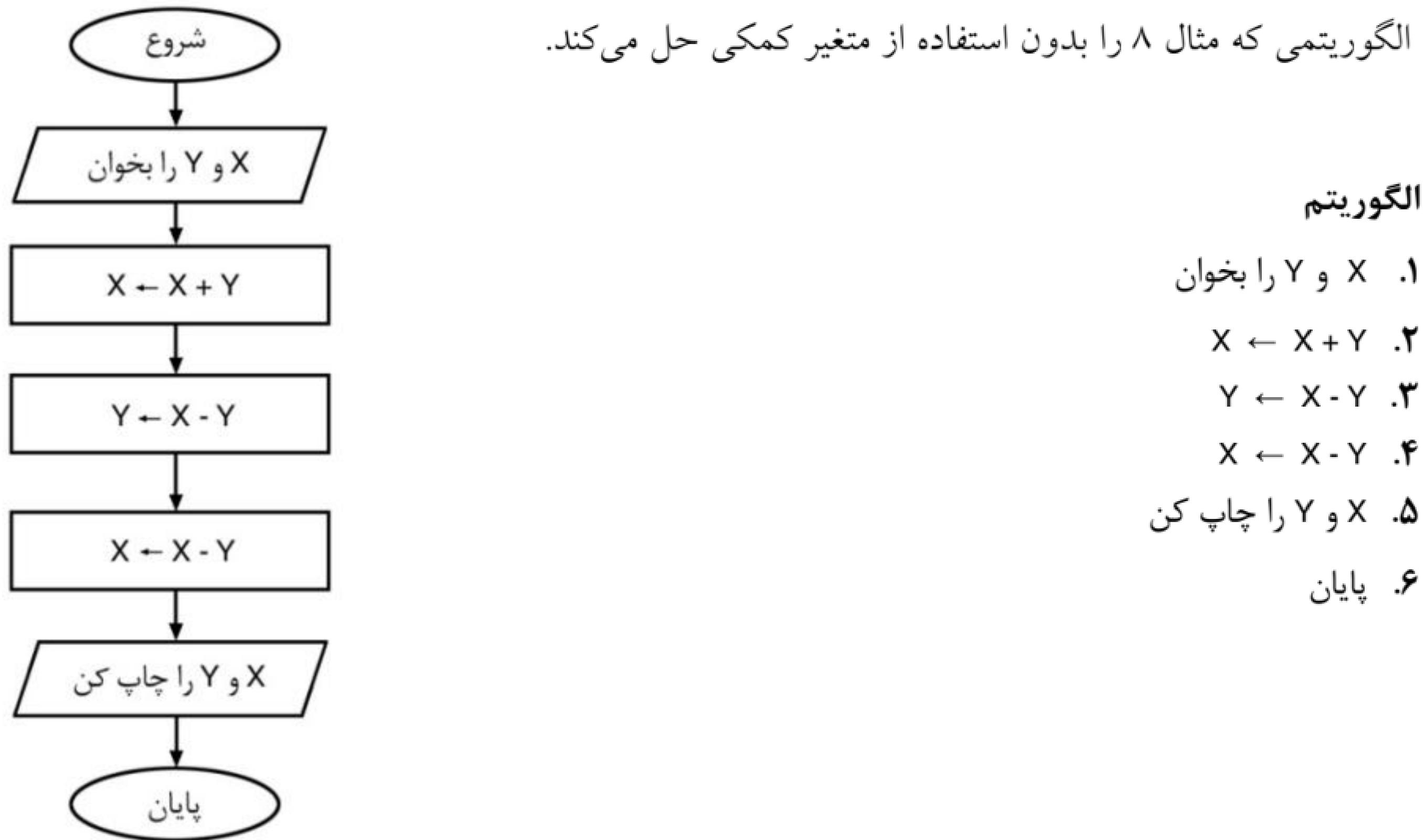
توضیح : برای حل این مسئله، یک متغیر کمکی به نام TEMP در نظر می‌گیریم. ابتدا مقدار X را به این متغیر منتقل کرده، مقدار Y را در X قرار می‌دهیم. اکنون مقدار قبلی X را که در TEMP قرار دارد، در Y قرار می‌دهیم.

الگوریتم

۱. X و Y را بخوان
۲. $TEMP \leftarrow X$
۳. $X \leftarrow Y$
۴. $Y \leftarrow TEMP$
۵. X و Y را چاپ کن
۶. پایان

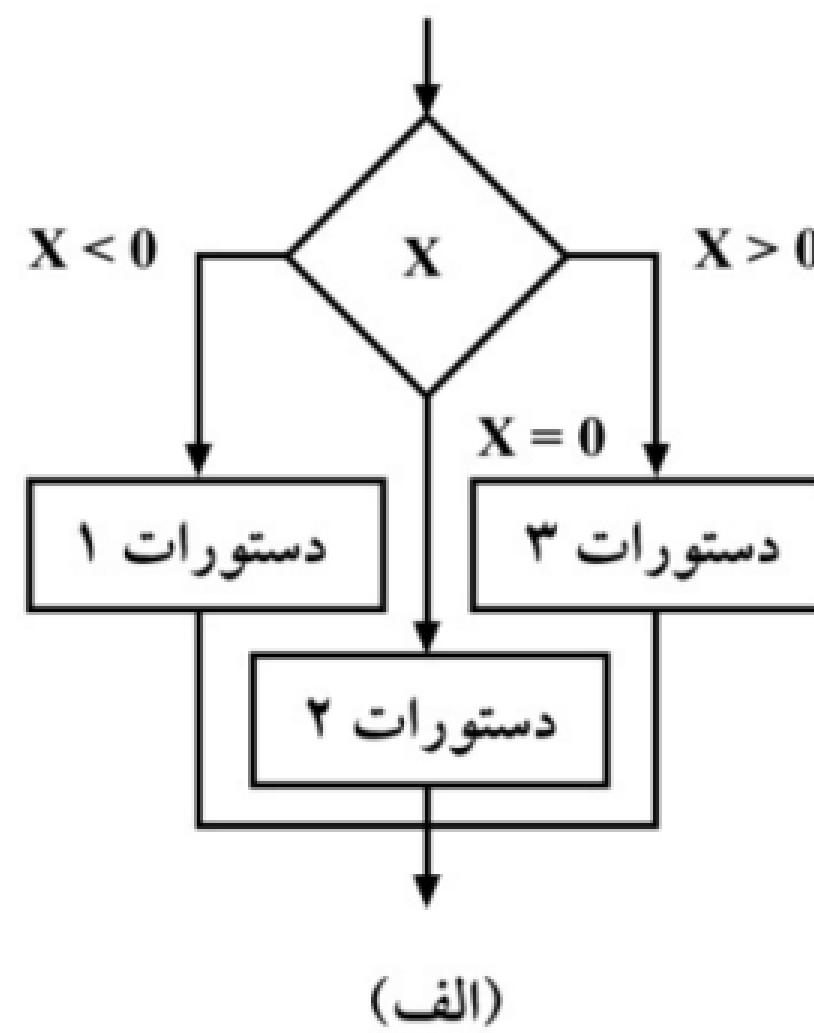
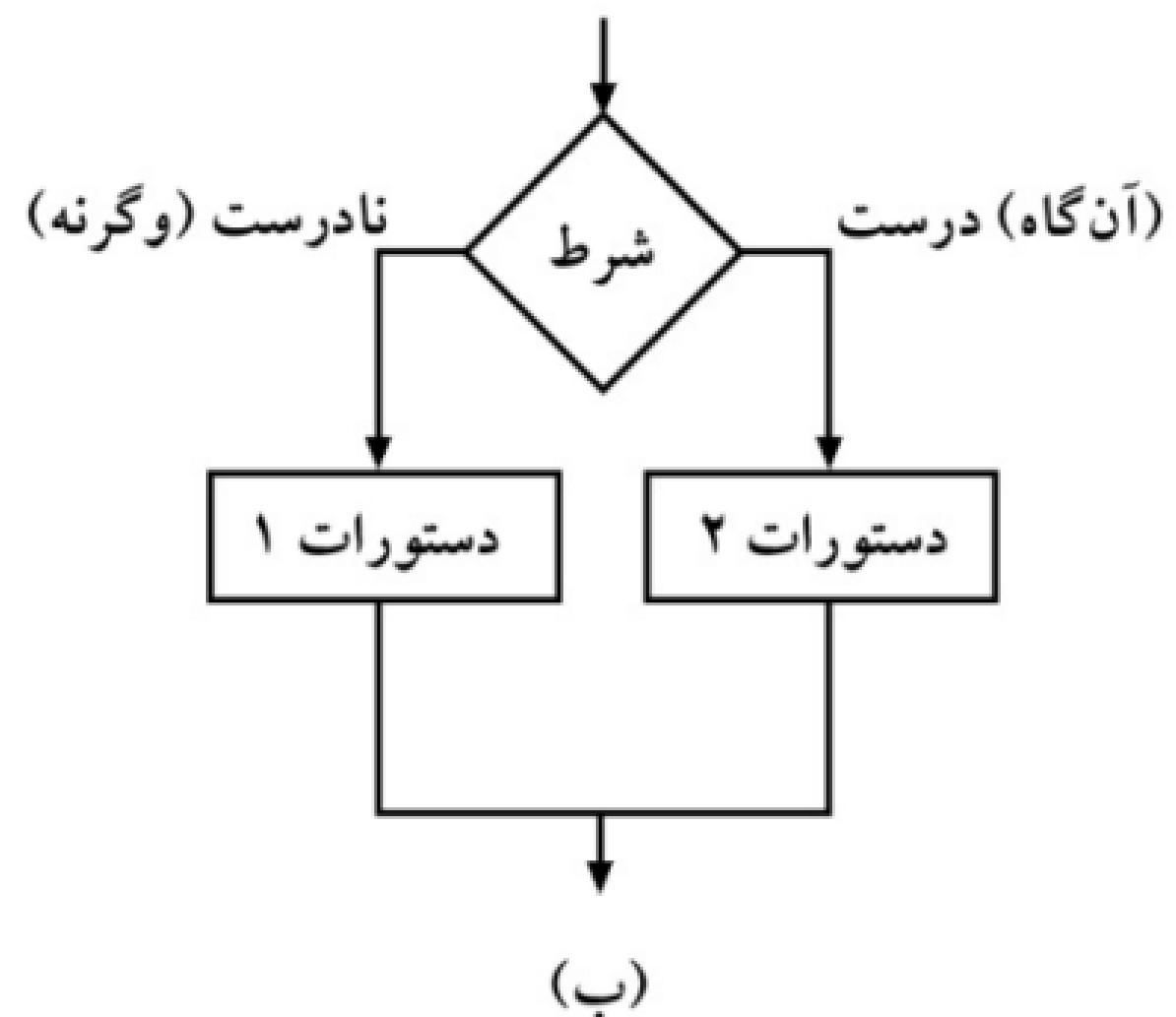
◀ مثال ۹-۳

الگوریتمی که مثال ۸ را بدون استفاده از متغیر کمکی حل می‌کند.

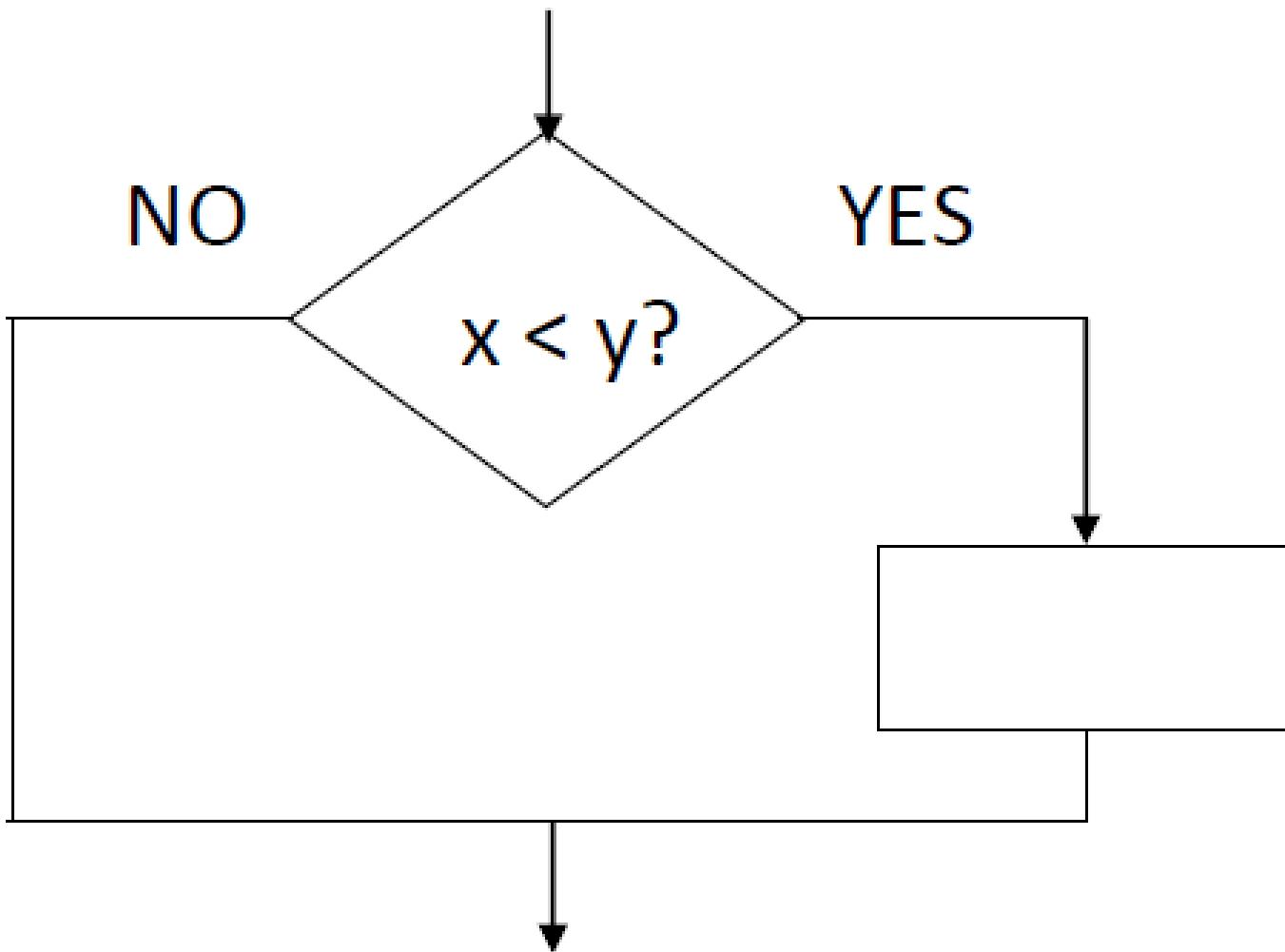


۱۰-۳ علامت‌های شرطی در فلوچارت

برای نمایش دستورات شرطی در فلوچارت، از لوزی استفاده می‌شود. لوزی می‌تواند دارای دو یا سه خروجی باشد. شکل (الف) با سه خروجی و شکل (ب) با دو خروجی است.



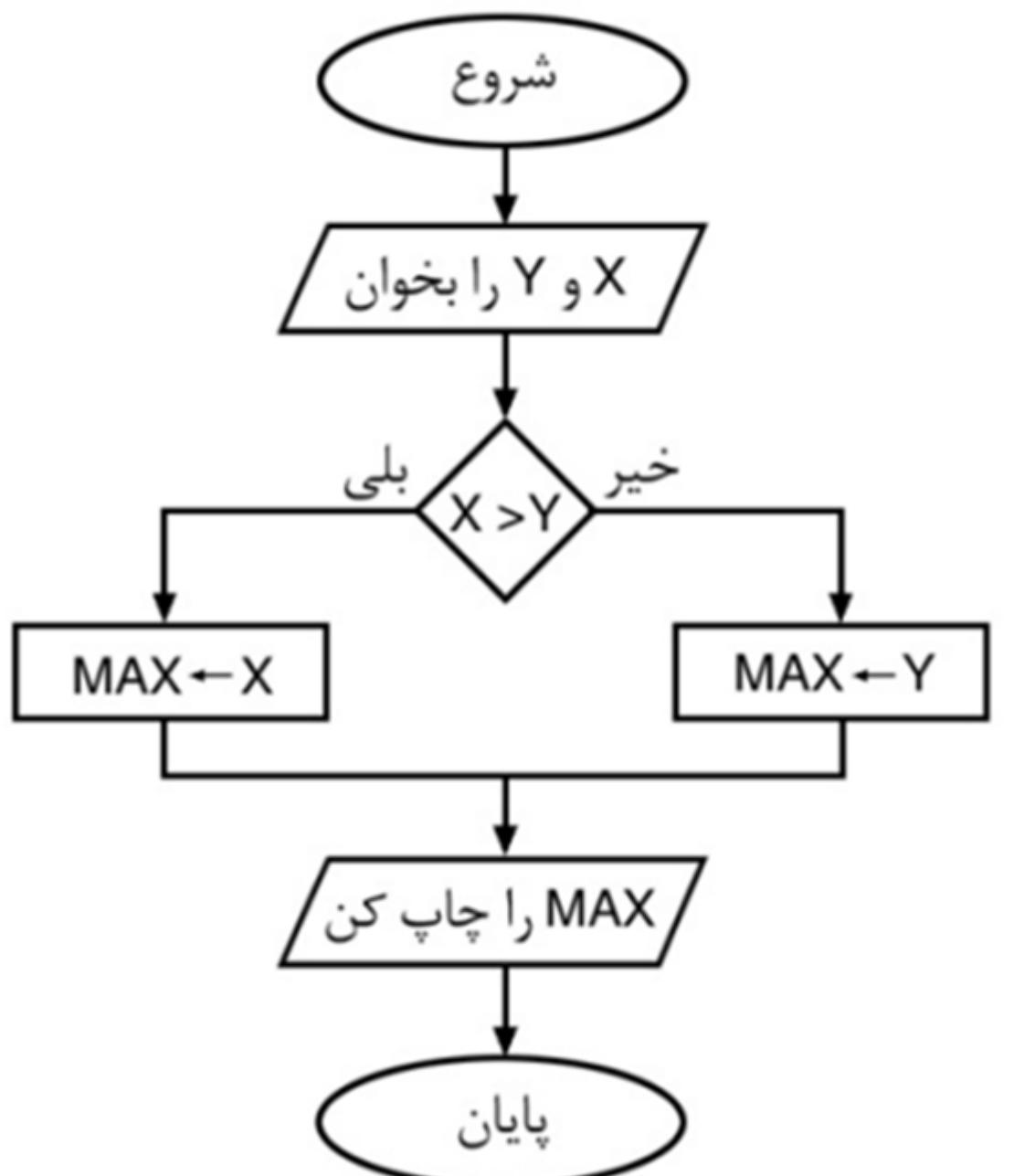
Flowchart



◀ مثال ۳-۱

الگوریتمی که دو مقدار را از ورودی خوانده، مقدار بزرگ‌تر را در خروجی چاپ می‌کند.

فلوچارت



متغیرها

X	مقدار اول
Y	مقدار دوم
MAX	مقدار بزرگ‌تر

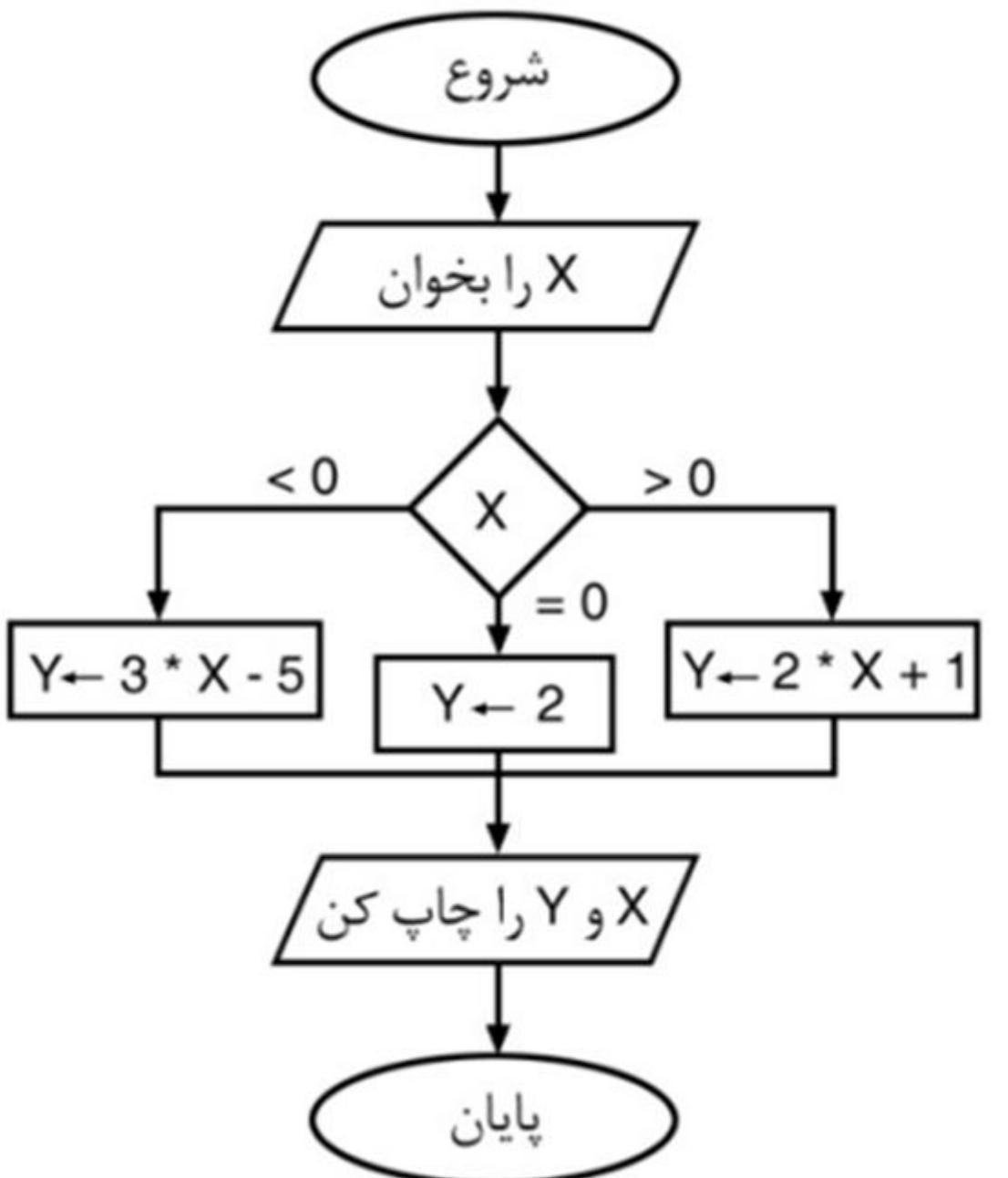
الگوریتم

۱. X و Y را بخوان
۲. اگر $Y > X$ آنگاه $MAX \leftarrow Y$ و گرنه $MAX \leftarrow X$
۳. MAX را چاپ کن
۴. پایان

الگوریتمی که عددی مثل X را از ورودی می‌خواند و مقدار Y را به صورت زیر محاسبه می‌کند:

$$\begin{array}{ll} Y = 3X - 5 & X < 0 \\ Y = 2 & X = 0 \\ Y = 2X + 1 & X > 0 \end{array}$$

فلوچارت



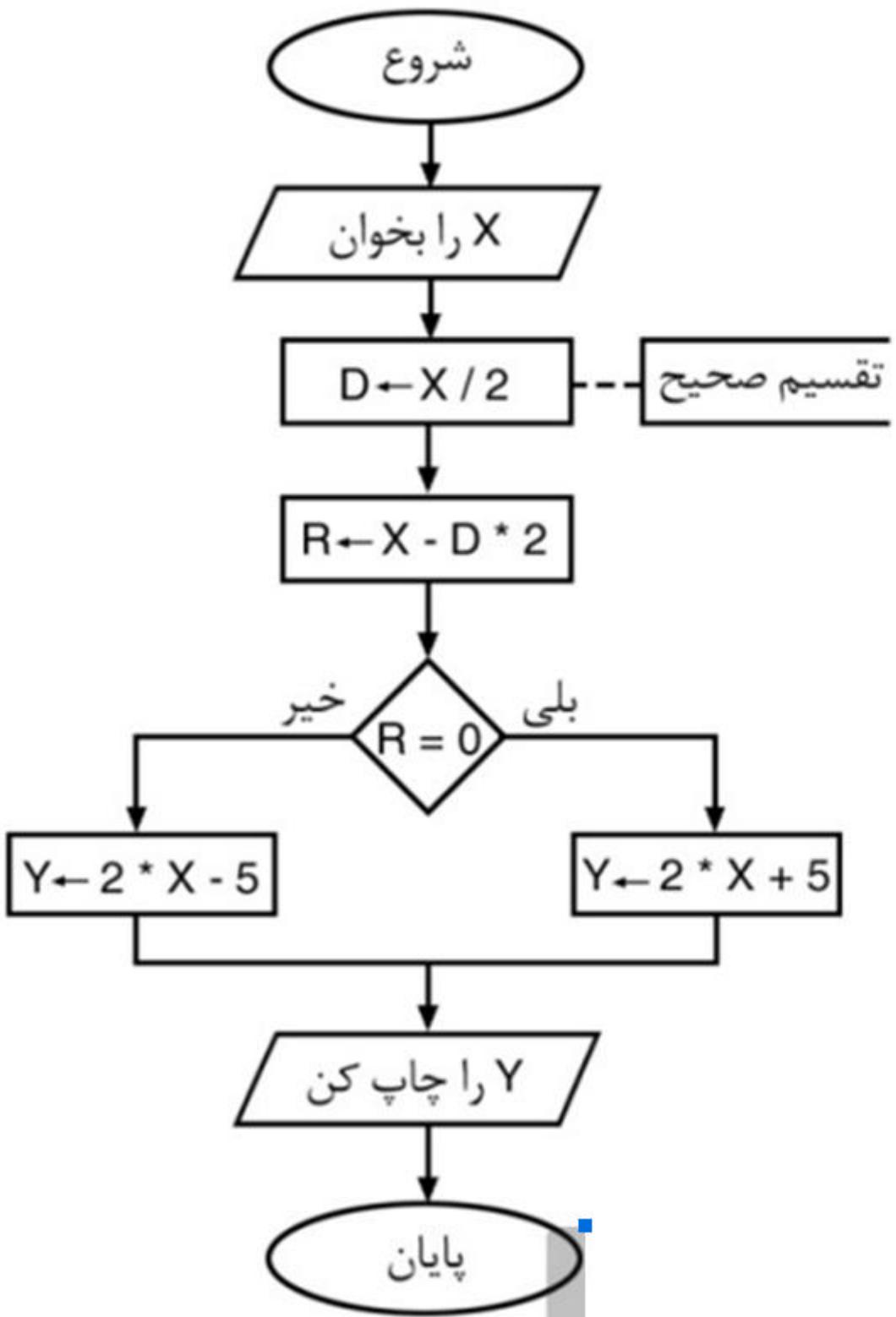
الگوریتم

۱. X را بخوان
۲. اگر $X < 0$ آن‌گاه $Y \leftarrow 3 * X - 5$
- وگرنه، اگر $X = 0$ آن‌گاه $Y \leftarrow 2$
- وگرنه $Y \leftarrow 2 * X + 1$
۳. X و Y را چاپ کن
۴. پایان

◀ مثال ۱۲-۳

الگوریتمی که عددی مثل X را از ورودی خوانده، اگر عدد زوج باشد، عبارت $2X + 5$ ولی اگر فرد باشد، عبارت $2X - 5$ را محاسبه می‌کند و به خروجی می‌برد.

فلوچارت



متغیرها

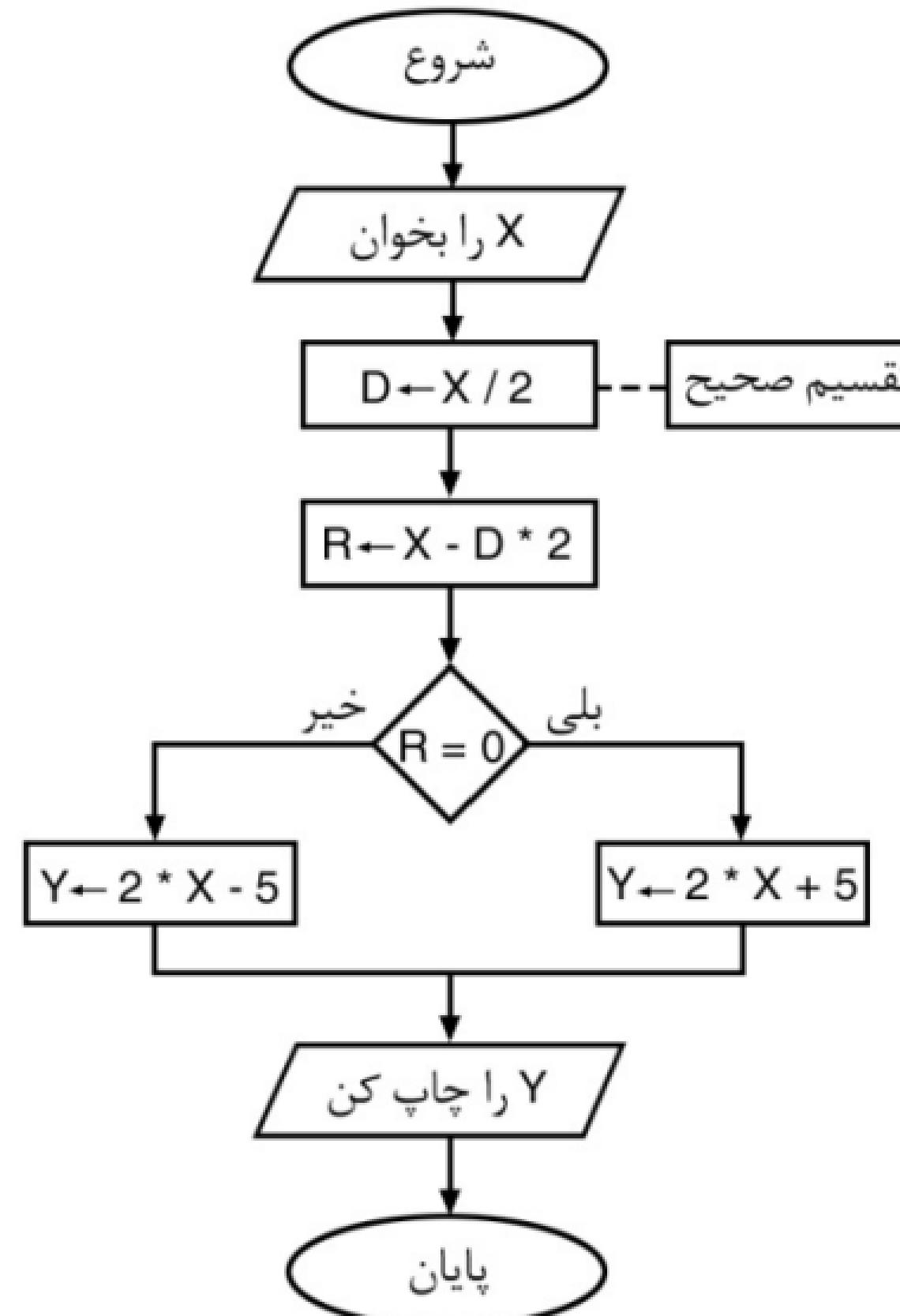
X	عدد مورد بررسی
Y	حاصل عبارت
R	باقیمانده تقسیم بر ۲
D	خارج قسمت تقسیم صحیح

الگوریتم

۱. X را بخوان
۲. $D \leftarrow [X / 2]$
۳. $R \leftarrow X - D * 2$
۴. اگر $R = 0$ آن‌گاه $Y \leftarrow 2 * X + 5$ وگرنه، $Y \leftarrow 2 * X - 5$
۵. Y را چاپ کن
۶. پایان

◀ مثال ۱۲-۳

الگوریتمی که عددی مثل X را از ورودی خوانده، اگر عدد زوج باشد، عبارت $2X + 5$ ولی اگر فرد باشد، عبارت $2X - 5$ را محاسبه می‌کند و به خروجی می‌برد.

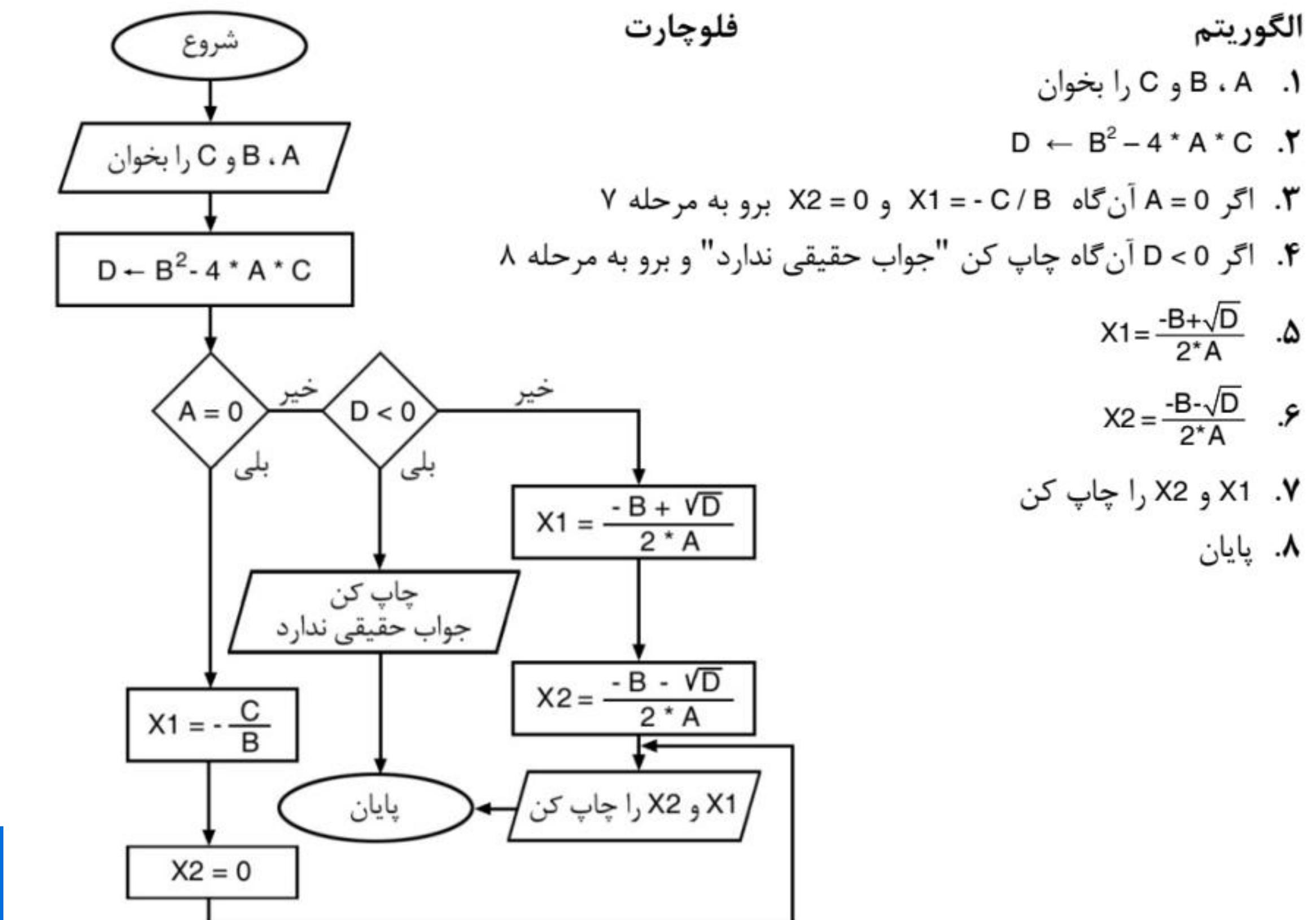


امتحان الگوریتم برای $X = 4$					
خروجی	Y	R	D	X	شماره دستور
				4	1
			2		2
		0			3
	13				4
13					5
					6

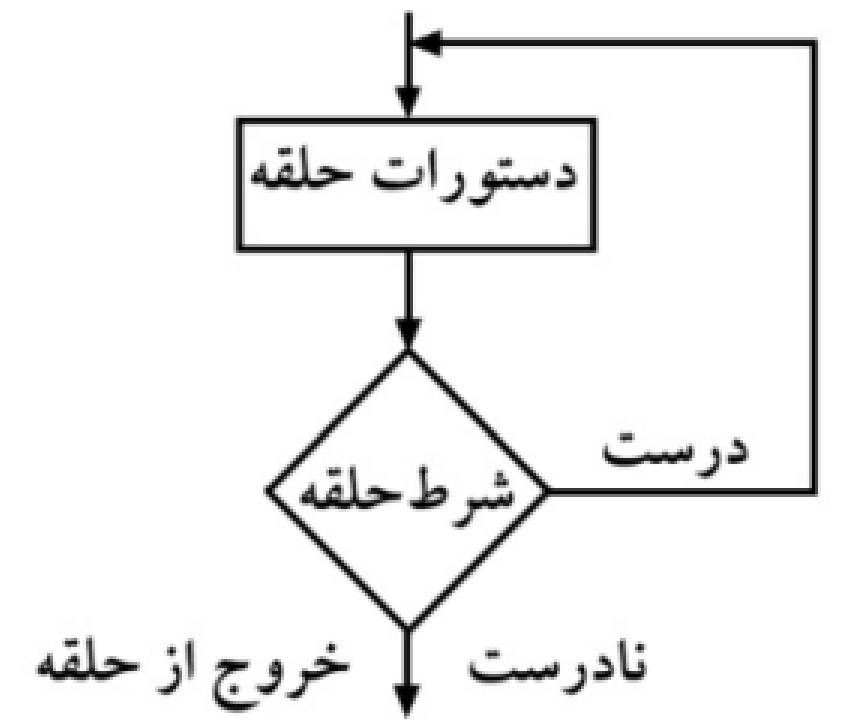
امتحان الگوریتم برای $X = 7$					
خروجی	Y	R	D	X	شماره دستور
				7	1
			3		2
	1				3
	9				4
9					5
					6

◀ مثال ۳-۱۵

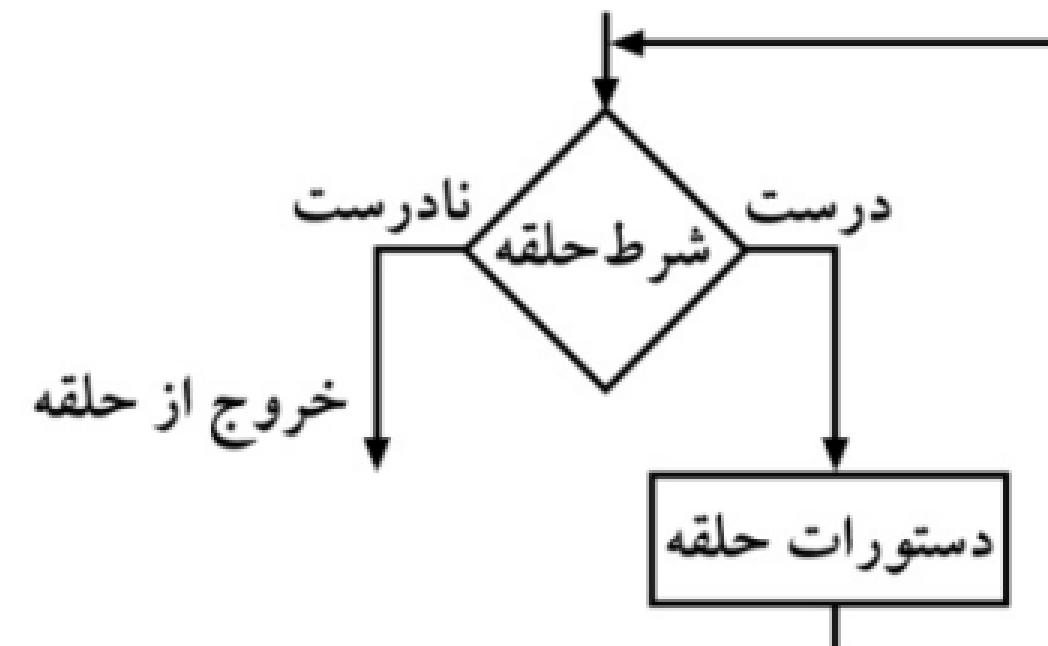
الگوریتمی که جواب‌های حقیقی معادله درجه دوم $AX^2 + BX + C = 0$ را پیدا کرده، در خروجی چاپ می‌کند.



۱۲-۳ حلقه‌های تکرار



(ب)

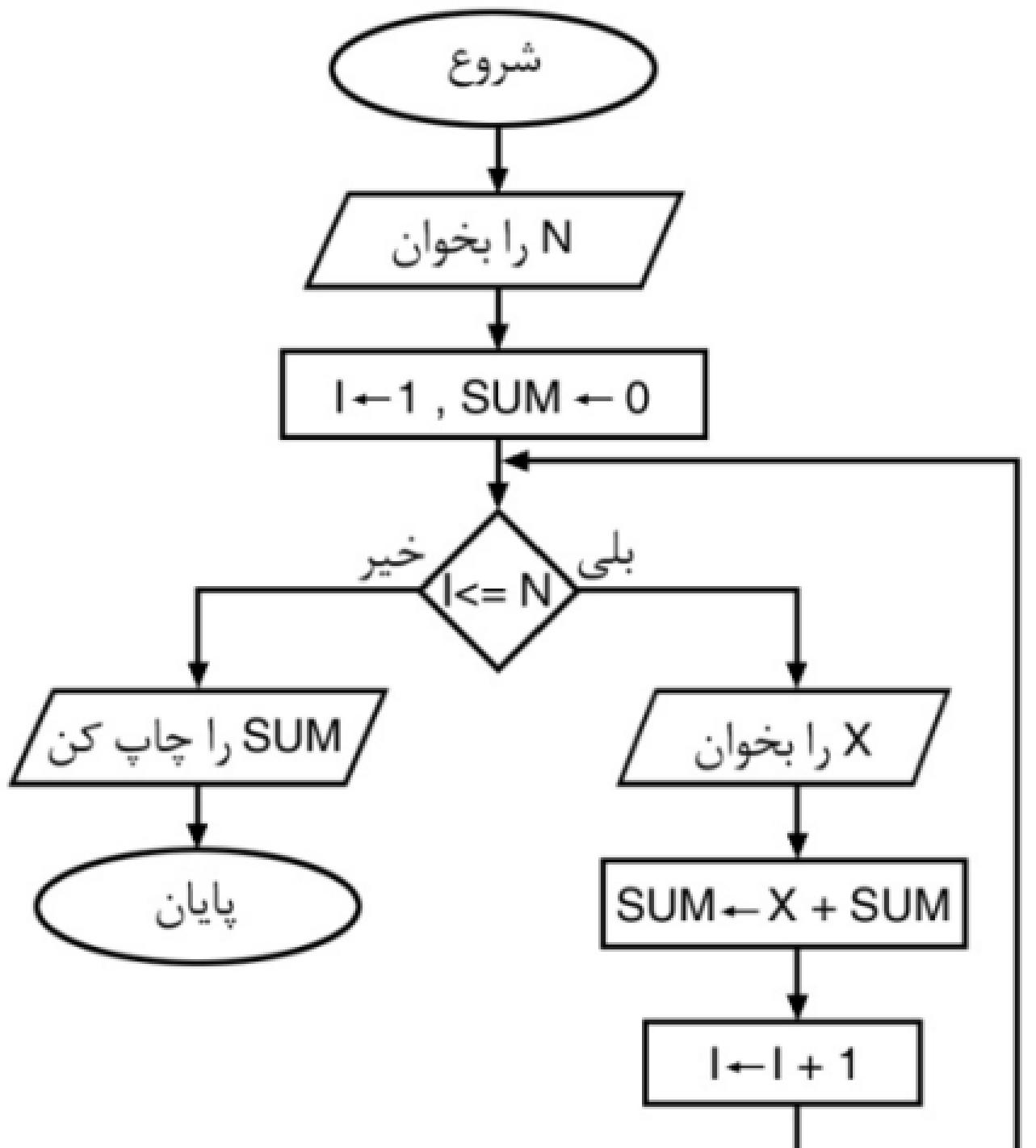


(الف)

◀ مثال ۳-۱۸

الگوریتمی که تعداد N عدد را از ورودی خوانده، مجموع آنها را محاسبه می‌کند و به خروجی می‌برد.

فلوچارت



متغیرها

X	عدد خوانده شده
I	شمارنده حلقه
SUM	مجموع اعداد

الگوریتم

۱. N را بخوان
۲. $SUM \leftarrow 0, I \leftarrow 1$
۳. تا زمانی که $I \leq N$ است، دستورات ۴ تا ۶ را تکرار کن
۴. X را بخوان
۵. $SUM \leftarrow X + SUM$
۶. $I \leftarrow I + 1$
۷. پایان حلقه
۸. SUM را چاپ کن
۹. پایان

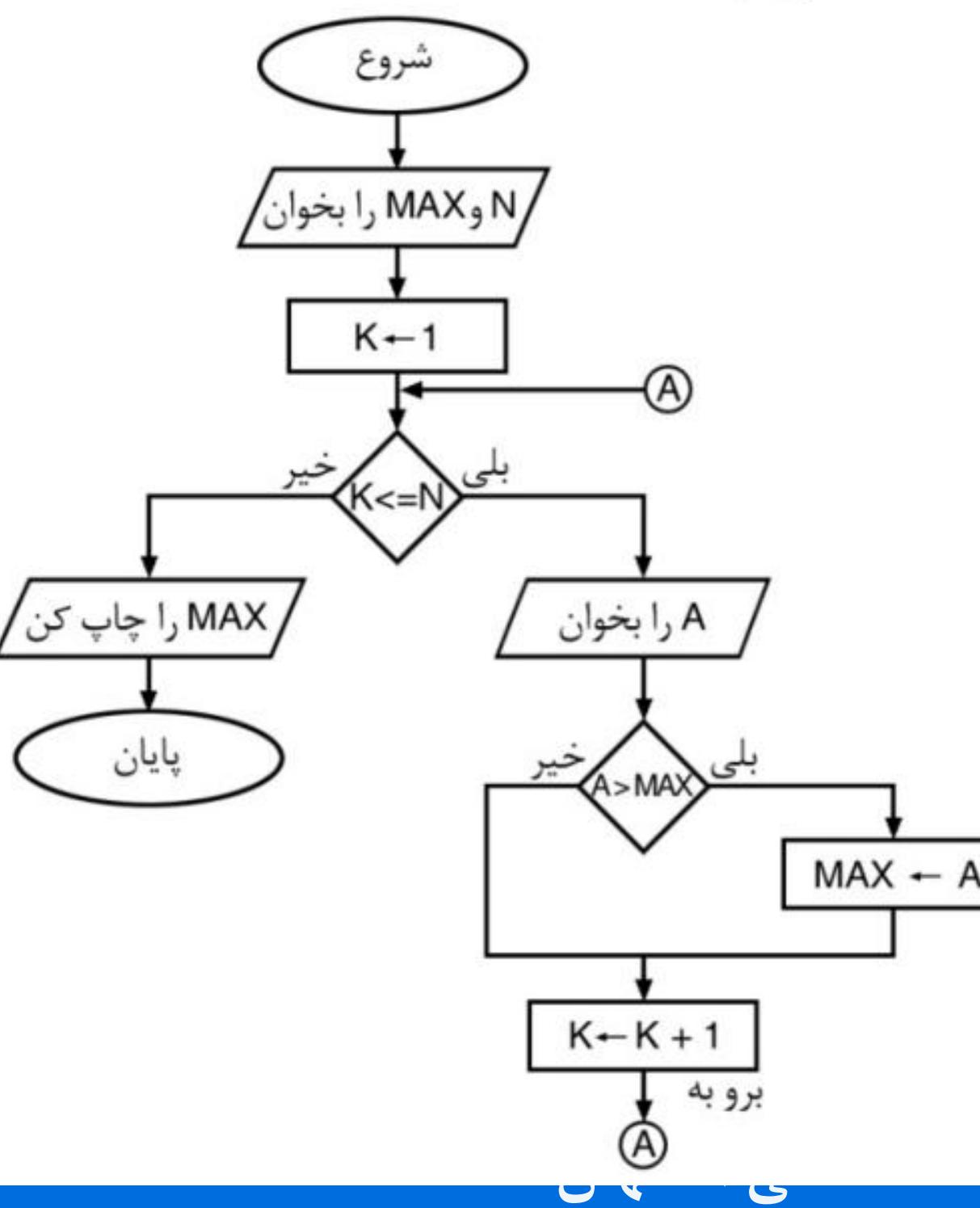
متغیرها

N	تعداد اعداد
MAX	بزرگ‌ترین عدد
K	شمارنده
A	عدد مورد بررسی

مثال ۳ ◀

الگوریتمی که تعدادی عدد را خوانده، بزرگ‌ترین عدد را پیدا و چاپ می‌کند.

فلوچارت



الگوریتم

۱. N را بخوان
۲. MAX را بخوان
۳. $K \leftarrow 1$
۴. تا زمانی که $K \leq N$ است، دستورات ۵ تا ۷ را تکرار کن
۵. A را بخوان
۶. اگر $A > MAX$ آنگاه $MAX \leftarrow A$
۷. $K \leftarrow K + 1$
۸. پایان حلقه
۹. MAX را چاپ کن
۱۰. پایان

الگوریتمی که عدد صحیح و مثبت N را از ورودی خوانده، فاکتوریل آن را محاسبه می‌کند.

متغیرها	
FACT	فاکتوریل
J	شمارنده
N	عدد موردنظر

توضیح

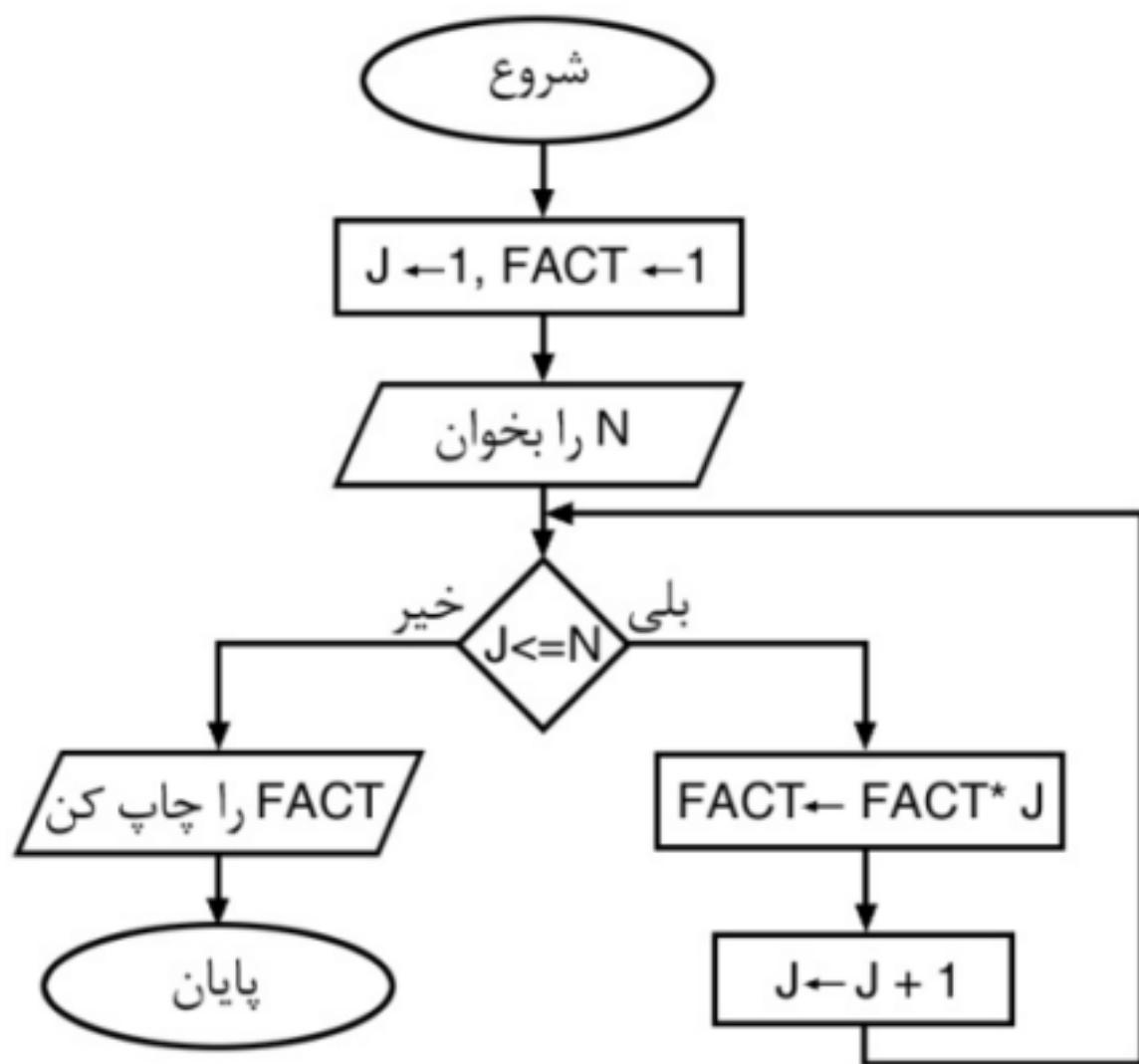
فاکتوریل عددی مثل N به صورت زیر محاسبه می‌شود:

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

به عنوان مثال، فاکتوریل عدد 4 به صورت زیر محاسبه می‌شود:

$$4! = 1 * 2 * 3 * 4$$

فلوچارت



الگوریتم

1. $J \leftarrow 1$

2. $FACT \leftarrow 1$

3. N را بخوان

4. تا زمانی که $N \leq J$ است،

دستورات ۵ تا ۶ را تکرار کن

5. $FACT \leftarrow FACT * J$

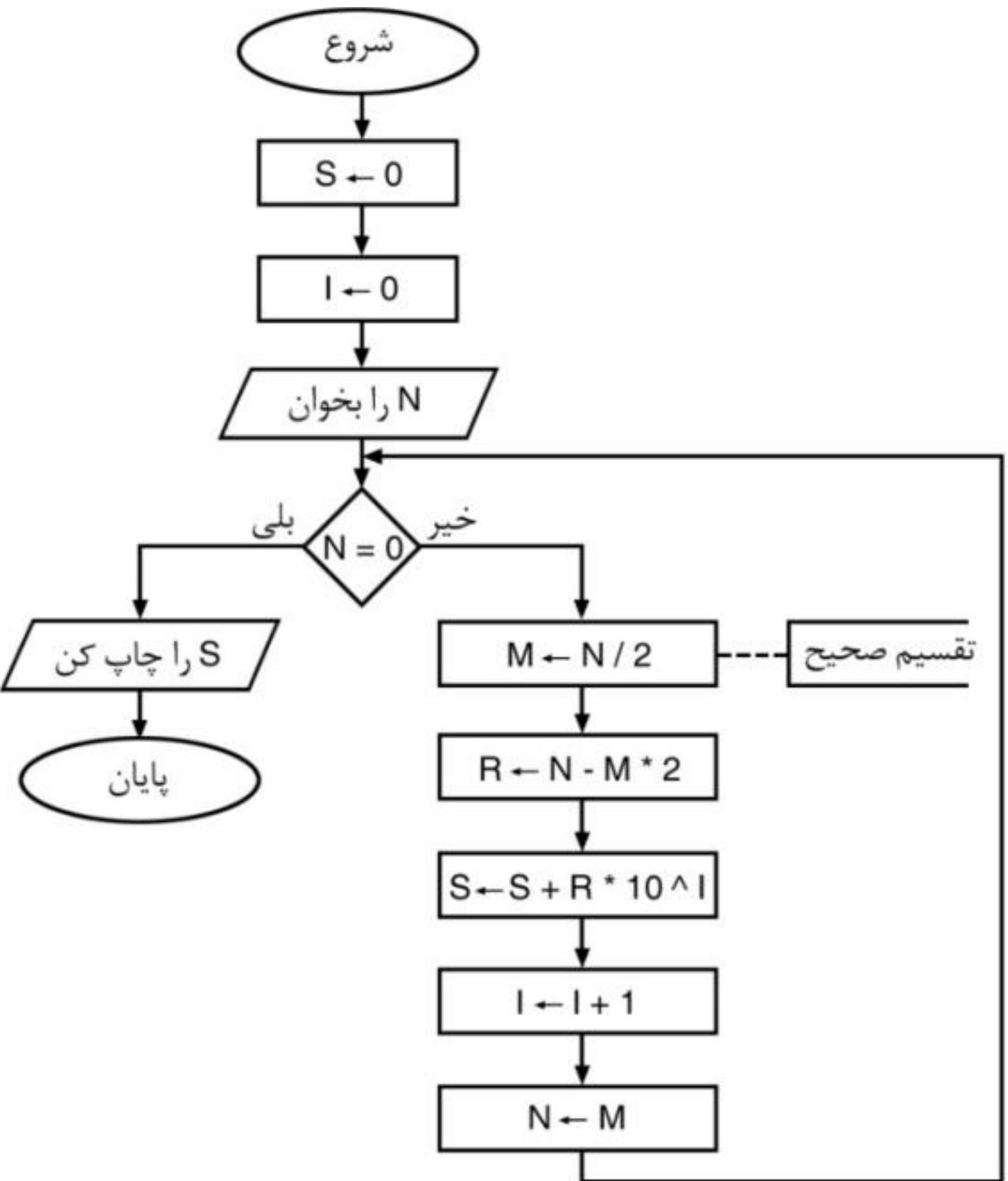
6. $J \leftarrow J + 1$

7. پایان حلقه

8. FACT را چاپ کن

9. پایان

الگوریتمی که یک عدد مبنای ۱۰ را خوانده، آن را به مبنای ۲ می‌برد و نتیجه را چاپ می‌کند.



۱. $S \leftarrow 0$

۲. $I \leftarrow 0$

۳. N را بخوان

۴. تا زمانی که $0 \neq N$ است،

دستورات ۵ تا ۹ را تکرار کن

۵. $M \leftarrow N / 2$

۶. $R \leftarrow N - M * 2$

۷. $S \leftarrow S + R * 10^I$

۸. $I \leftarrow I + 1$

۹. $N \leftarrow M$

۱۰. پایان حلقه

۱۱. S را چاپ کن

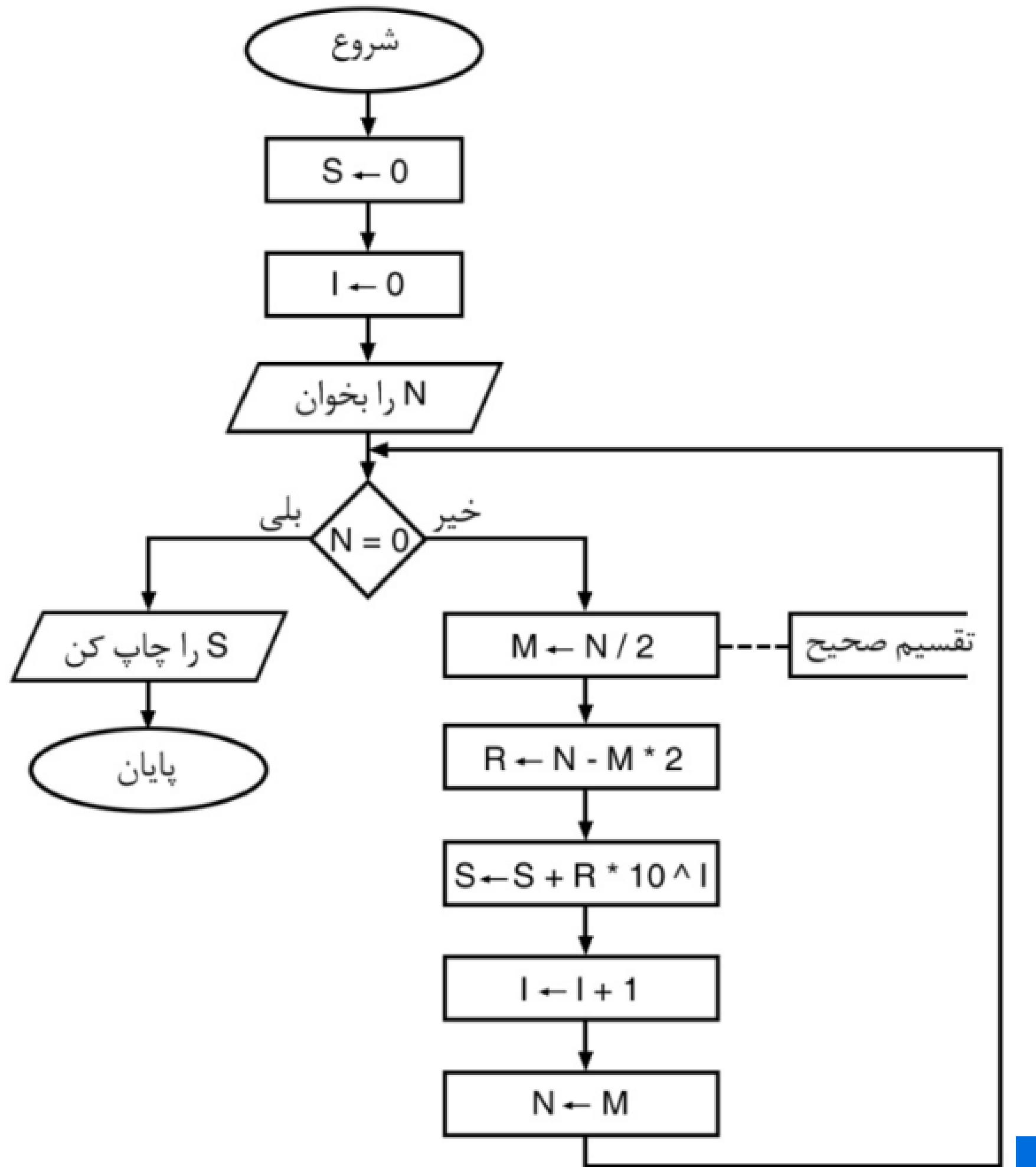
۱۲. پایان

وقتی الگوریتم را به ازای

$N = 3$ امتحان کردیم، مقدار

۱۱ در S قرار گرفت که

نتیجه‌ی مورد انتظار ماست.

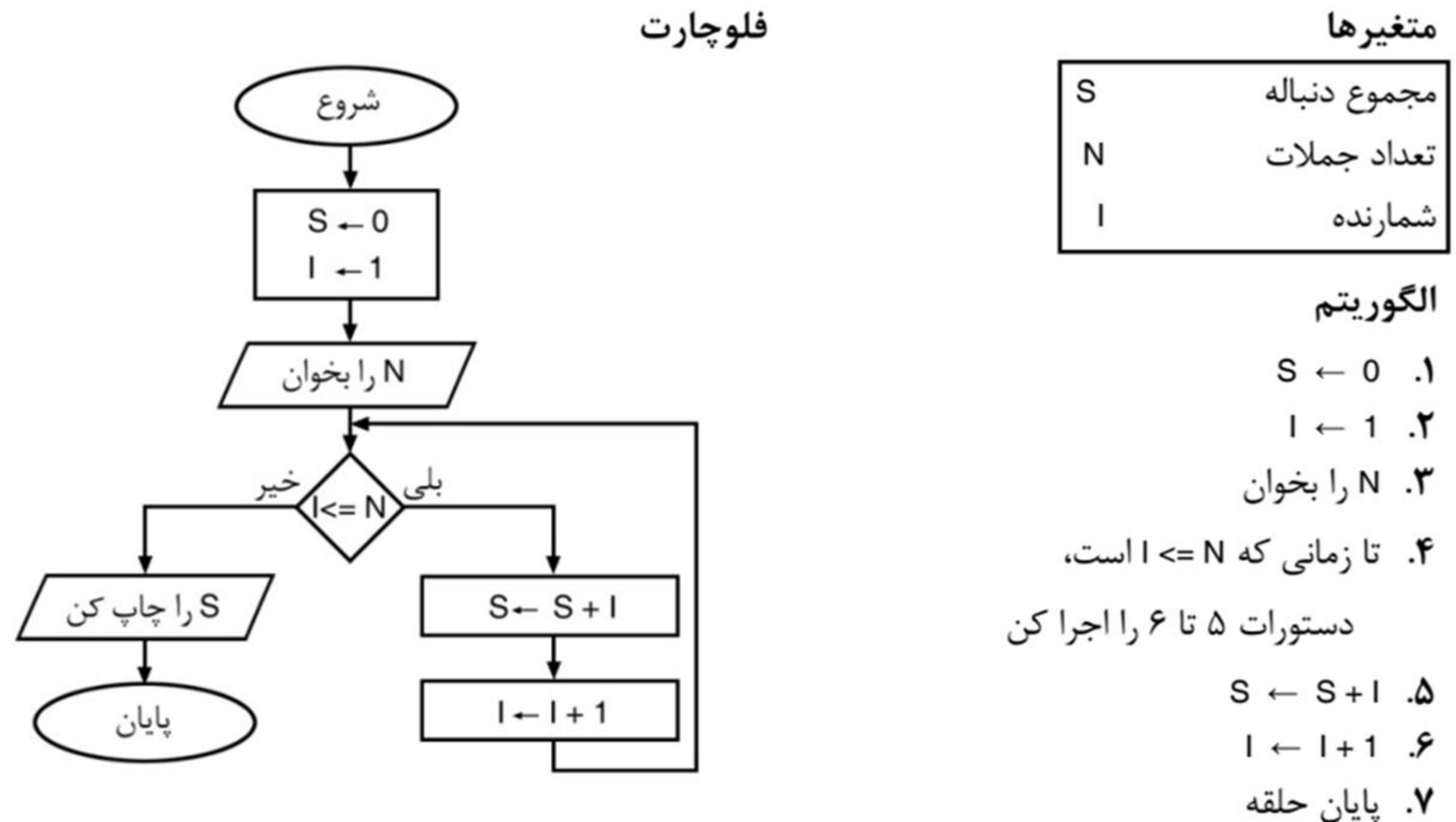


S	R	M	I	N	شماره دستور
0					1
			0		2
				3	3
					4
		1			5
	1				6
1					7
			1		8
				1	9
					10
					4
			0		5
		1			6
11					7
			2		8
				0	9

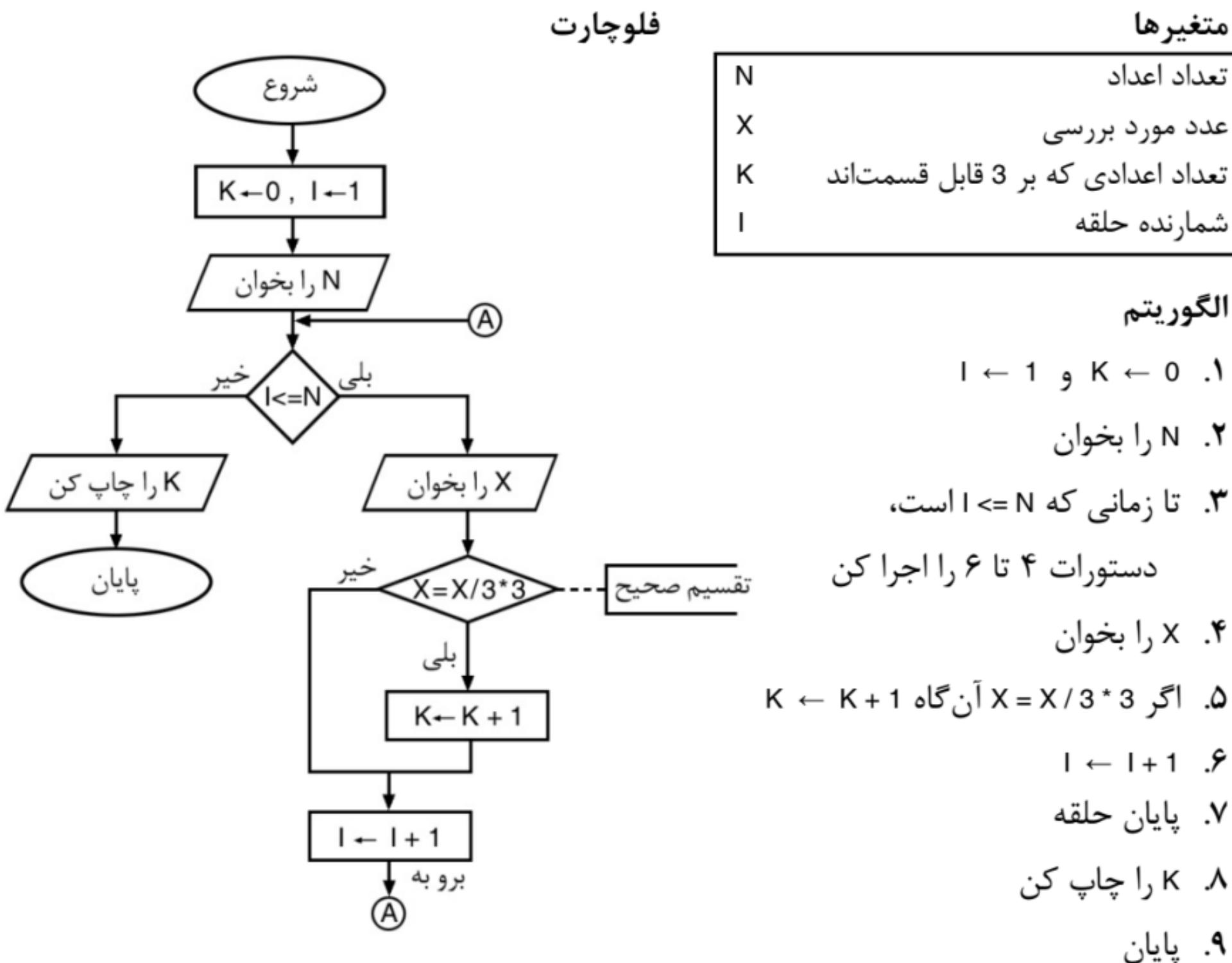
الگوریتمی که مجموع N جمله از دنباله‌ی زیر را محاسبه کرده، به خروجی می‌برد.

$$S = 1 + 2 + 3 + \dots + N$$

توضیح: برای مجموع جملات این دنباله، کافی است اعداد ۱ تا N را تولید کرده، با هم جمع کنیم.



الگوریتمی که تعداد N عدد صحیح و مثبت را از ورودی خوانده، تعداد اعدادی را مشخص کند که بر ۳ قابل قسمت هستند.

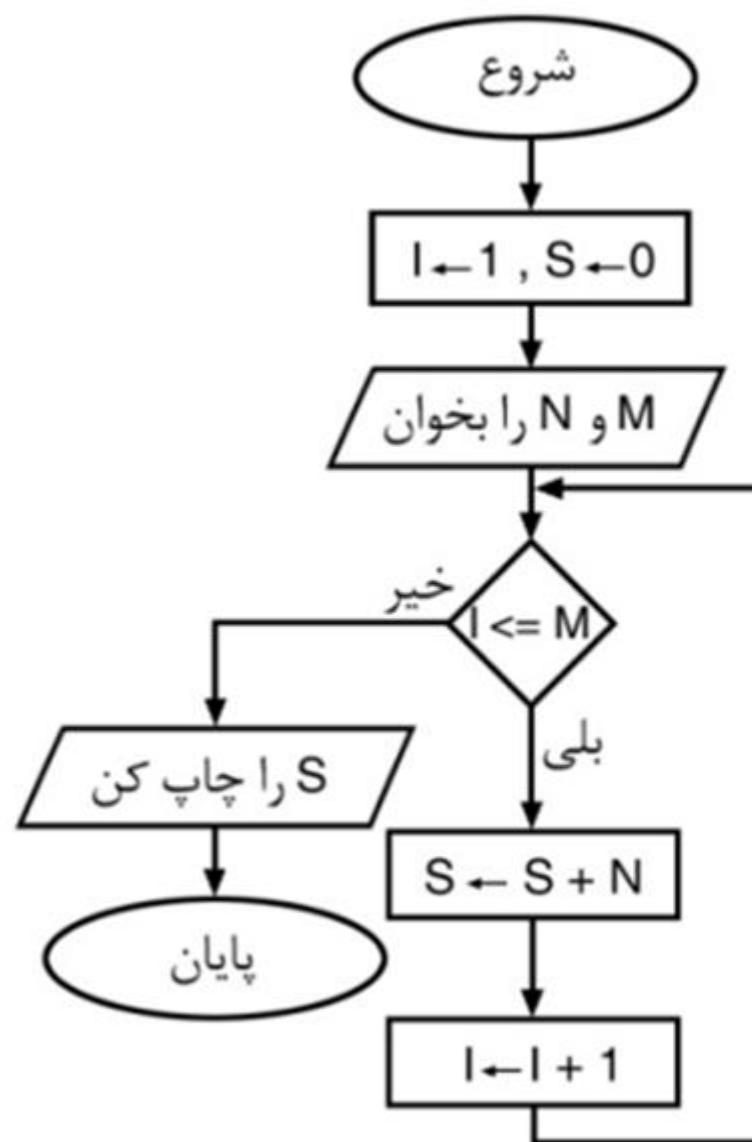


الگوریتمی که دو عدد صحیح مثبت را از ورودی خوانده، حاصل ضرب آنها را با استفاده از عمل جمع محاسبه می‌کند.

توضیح

برای ضرب دو عدد از طریق عمل جمع، باید یک عدد را به تعداد دفعاتی که معادل عدد دیگر است، با خودش جمع کرد. به عنوان مثال، برای ضرب دو عدد M و N می‌توان N را به مقدار M بار با خودش جمع کرد.

فلوچارت



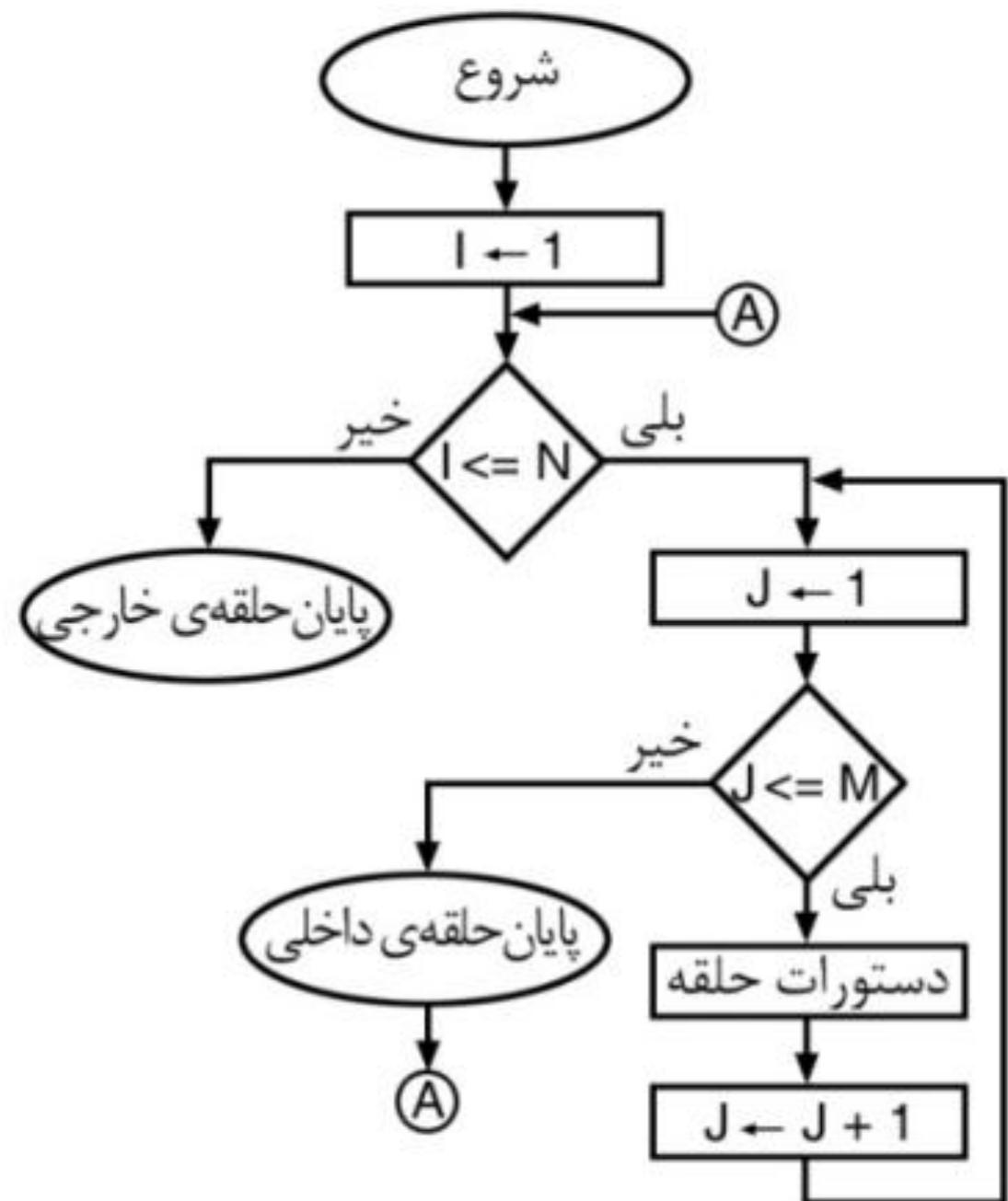
متغیرها

I	شمارنده
N, M	دو عدد صحیح
S	حاصل ضرب دو عدد

الگوریتم

۱. $S \leftarrow 0$
۲. $I \leftarrow 1$
۳. M و N را بخوان
۴. تا زمانی که $I \leq M$ است، دستورات ۵ تا ۶ را اجرا کن
۵. $S \leftarrow S + N$
۶. $I \leftarrow I + 1$
۷. پایان حلقه
۸. S را چاپ کن
۹. پایان

۱۳-۳ حلقه‌های تکرار تودر تو



حلقه‌های تودر تو

۵. به ازای $N \Rightarrow$ ادستورات ۶ تا ۱۲ را اجرا کن

.۶

.۷ $I \leftarrow 1$

۸. به ازای $M \Rightarrow$ لدستورات ۹ تا ۱۰ را اجرا کن

.۹

.۱۰ $J \leftarrow J + 1$

.۱۱. پایان حلقه $M \Leftarrow J$

.۱۲ $I \leftarrow I + 1$

.۱۳. پایان حلقه $I \Leftarrow N$

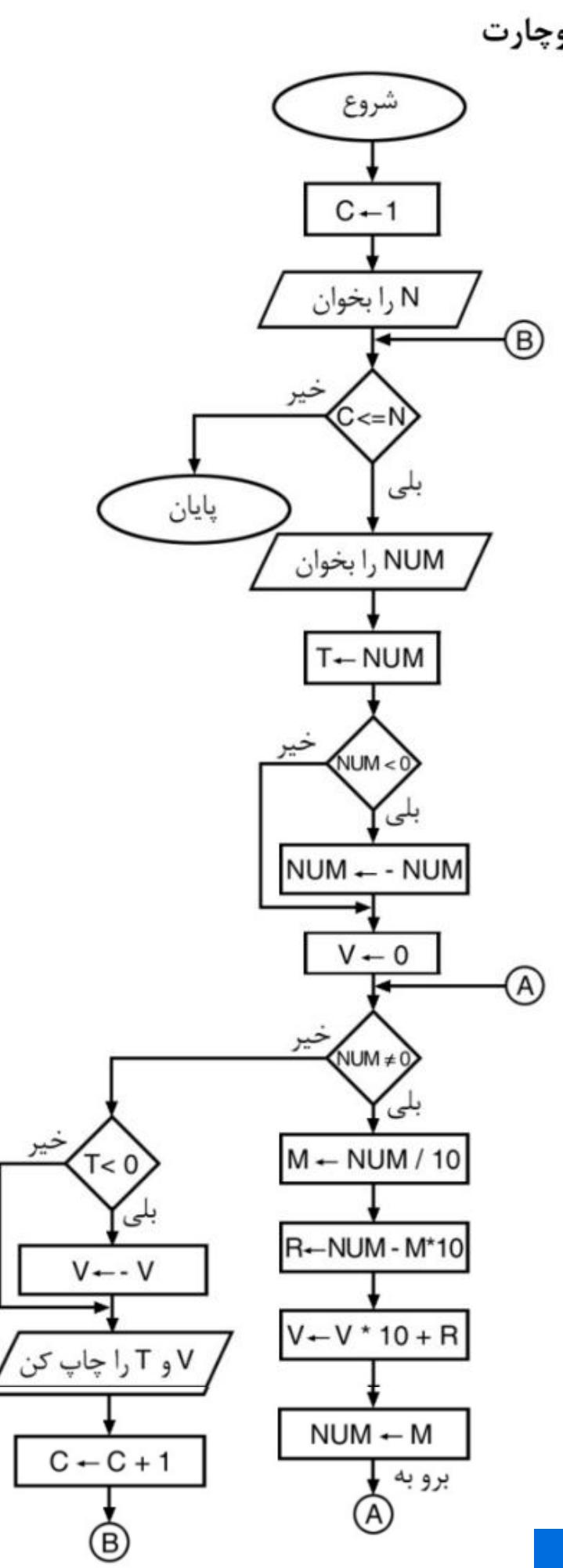
.۱۴. پایان

◀ مثال ۳۷-۳

الگوریتمی که تعدادی عدد صحیح را خوانده، وارون آنها را محاسبه و چاپ کند.

توضیح

وارون عددی مثل ۴۲۷ عدد ۷۲۴ است. الگوریتم، عددی را خوانده در صورتی که آن عدد منفی باشد، آن را به عدد مثبت تبدیل می‌کند. سپس باقیمانده‌ی تقسیم آن عدد بر ۱۰ را محاسبه می‌کند. خارج قسمتِ تقسیم، به عنوان مقسوم‌علیه تقسیم جدید مورد استفاده قرار می‌گیرد و این روند ادامه می‌یابد تا عدد به صفر تبدیل شود. چون باقیمانده‌های تقسیم باید در کنار هم قرار گیرند، وارون عدد هر بار در ۱۰ ضرب شده با باقیمانده‌ی حاصل جمع می‌شود. متغیر T به این علت استفاده شد که مقدار اولیه‌ی عدد نگهداری شود تا بعداً چاپ گردد. در این الگوریتم دو حلقه‌ی تکرار وجود دارد، یکی از دستور ۳ تا ۱۸ و دیگری از دستور ۸ تا ۱۳. در حلقه‌ی بیرونی (۳ تا ۱۸)، عدد خوانده می‌شود. و در حلقه درونی (۸ تا ۱۳) وارون عدد حساب می‌شود.



الگوريتم

١. شروع
٢. $C \leftarrow 1$
٣. N را بخ
٤. تازمانی

۴. تا زمانی که $C \leq N$ است،

دستورات ۵ تا ۱۷ را اجرا کن

۵. NUM را بخوان

T \leftarrow NUM .9

۷. اگر $0 < \text{NUM} < \text{NUM}$ آن‌گاه $\text{NUM} \leftarrow -\text{NUM}$

$V \leftarrow 0$. \wedge

۹. تا زمانی که $NUM \neq 0$ است،

دستورات ۱۰ تا ۱۳ را اجرا کن

M \leftarrow NUM / 10 .1.

R \leftarrow NUM - M * 10 .11

$V \leftarrow V * 10 + R .12$

NUM \leftarrow M.13

۱۴. پایان حلقه (شروع از مرحله‌ی ۹)

۱۵. اگر $T < 0$ باشد، آن‌گاه $V \leftarrow -V$

۱۶. و T را چاپ کن

$C \leftarrow C + 1$

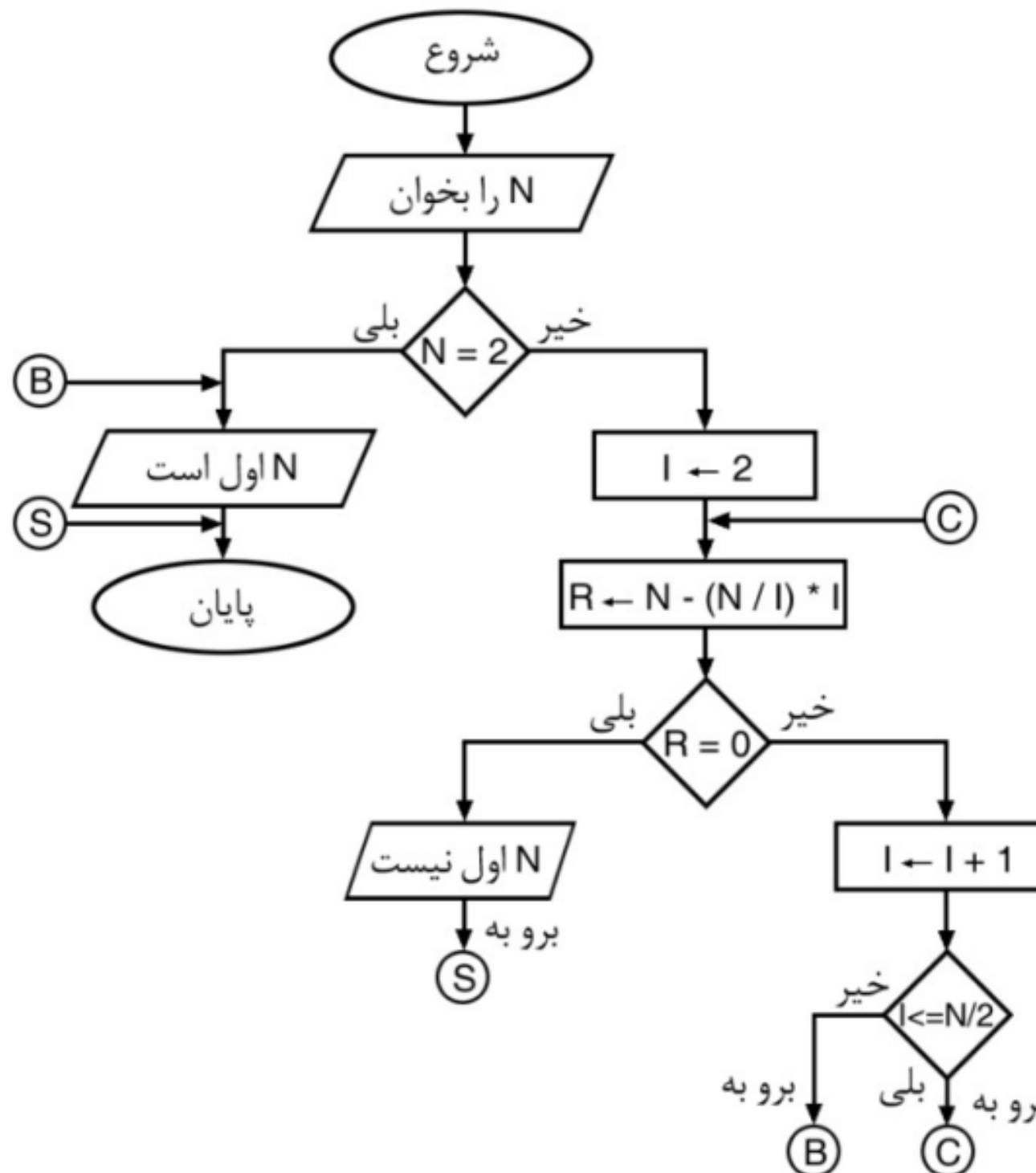
۱۸. پایان حلقه (شروع از مرحله‌ی ۴)

۱۹ پایان

◀ مثال ۳-۱

الگوریتمی که عددی مثل N را از ورودی خوانده، مشخص می‌کند که N اول است یا خیر.

فلوچارت



متغیرها

I	شمارنده
N	عدد مورد بررسی
R	باقیمانده تقسیم

الگوریتم

۱. شروع
۲. N را بخوان
۳. اگر N = 2 برو به مرحله ۹
۴. I ← 2
۵. R ← N - (N / I) * I (تقسیم صحیح)
۶. اگر R = 0 "N" اول نیست ، برو به مرحله ۱۰
۷. I ← I + 1
۸. اگر 2 ≤ I < N است ، برو به مرحله ۵
۹. چاپ کن "N اول است"
۱۰. پایان

◀ مثال ۳-۴

الگوریتمی که دو عدد صحیح و مثبت N را از ورودی خوانده، بزرگ‌ترین مقسوم‌علیه مشترک (ب.م.م) و کوچک‌ترین مضرب مشترک (ک.م.م) آن دو عدد را محاسبه می‌کند.

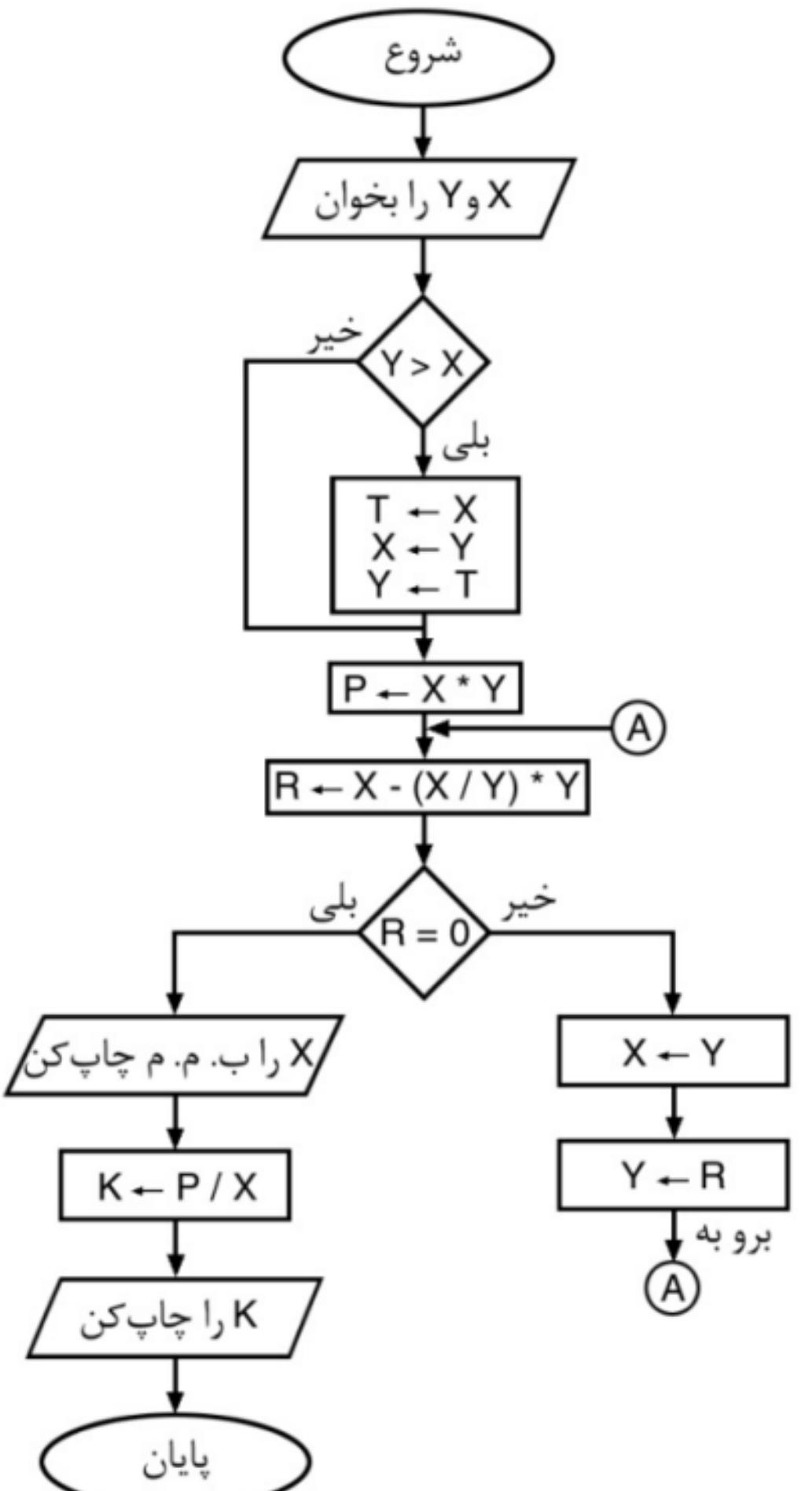
توضیح

برای به دست آوردن ب.م.م به این صورت عمل می‌کنیم: عدد بزرگ‌تر را بر عدد کوچک‌تر تقسیم می‌کنیم. اگر باقیمانده صفر بود، عدد کوچک‌تر (مقسوم‌علیه) همان ب.م.م است. وگرنه، مقسوم‌علیه را به عنوان مقسوم و باقیمانده را به عنوان مقسوم‌علیه محسوب کرده، عمل تقسیم را انجام می‌دهیم. این روند را آنقدر ادامه می‌دهیم تا باقیمانده صفر شود. در این صورت مقسوم‌علیه، ب.م.م است. می‌دانیم که رابطه بین ب.م.م و ک.م.م دو عدد به صورت زیر است.

$$\text{ب.م.م} / (\text{حاصلضرب دو عدد}) = \text{ک.م.م}$$

در این الگوریتم دو عدد X و Y را از ورودی می‌خوانیم X باید بزرگ‌تر از Y باشد، در غیر این صورت در الگوریتم، با استفاده از متغیر کمکی T جای آنها را عوض می‌کنیم.

فلوچارت



متغیرها

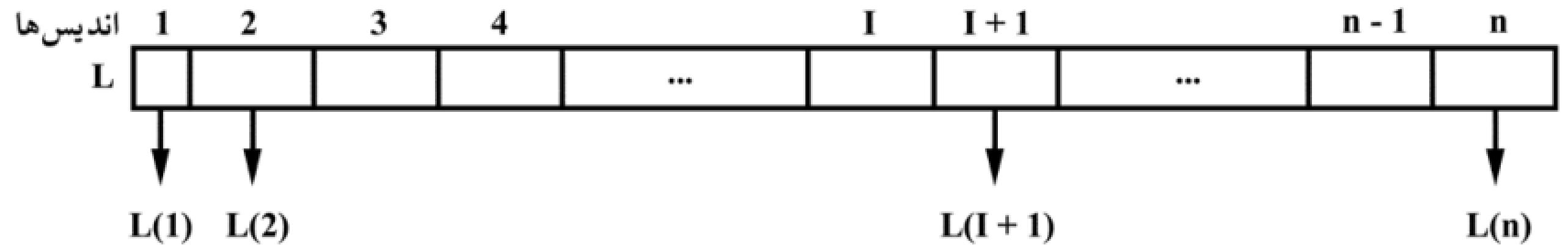
X, Y	دو عدد صحیح و مثبت
T	متغیر کمکی
R	باقیمانده تقسیم
K	ک. م. م
P	حاصل ضرب X در Y

الگوریتم

۱. X و Y را بخوان
۲. اگر $X > Y$ باشد، آن‌گاه

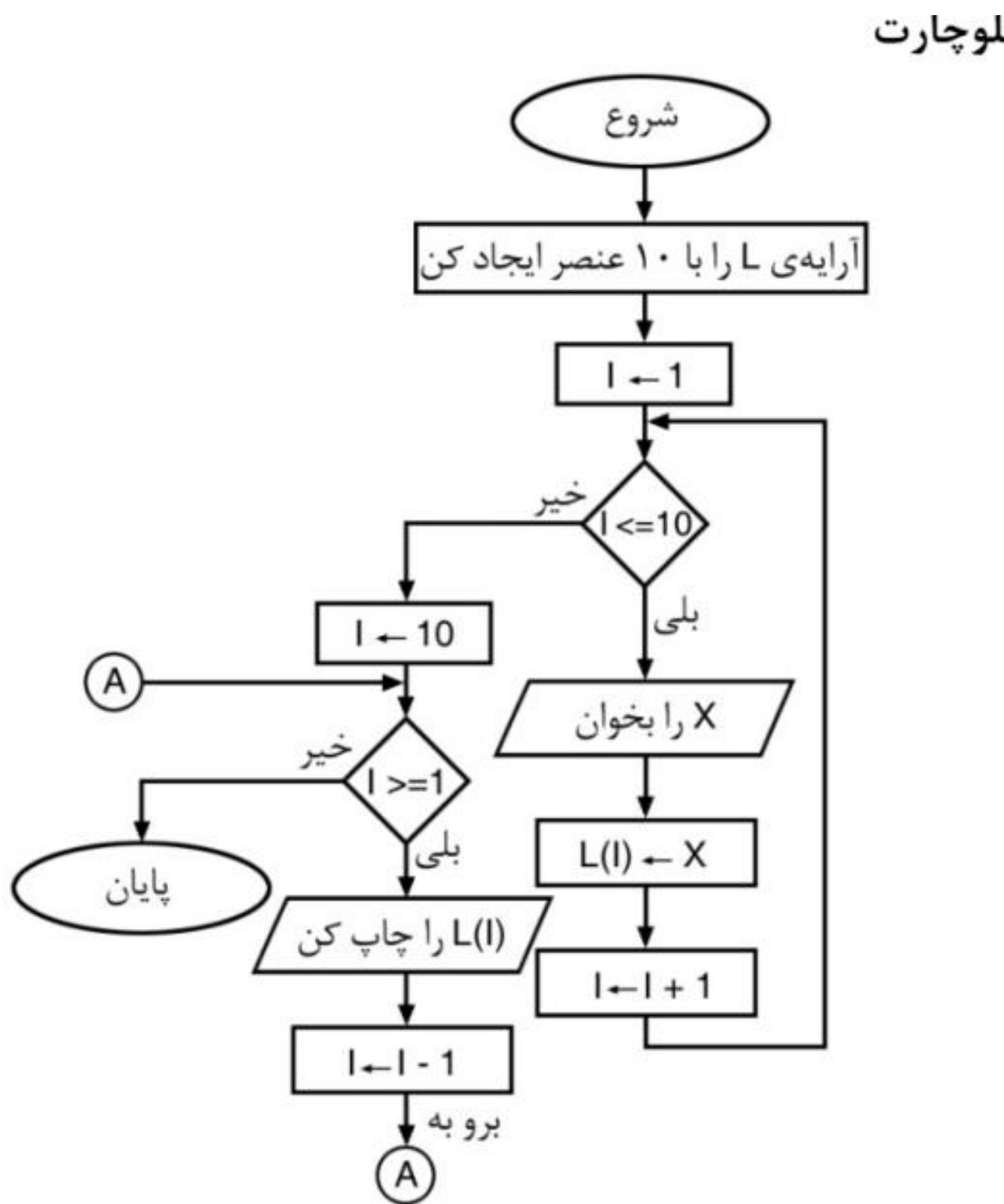
$$\begin{aligned}T &\leftarrow X \\ X &\leftarrow Y \\ Y &\leftarrow T\end{aligned}$$
۳. $P \leftarrow X * Y$
۴. $R \leftarrow X - (X / Y) * Y$
۵. اگر $R = 0$ برو به مرحله ۹
۶. $X \leftarrow Y$
۷. $Y \leftarrow R$
۸. برو به مرحله ۴
۹. X را به عنوان ب. م. م چاپ کن
۱۰. $K \leftarrow P / X$
۱۱. K را به عنوان ک. م. م چاپ کن
۱۲. پایان

کاربرد آرایه در الگوریتم



شکل ۴-۱ ▶ نمونه‌ای از آرایه

الگوریتمی که تعداد ۱۰ عدد را خوانده در آرایه‌ای قرار می‌دهد. سپس عناصر آرایه را از آخرین عنصر به اولین عنصر به خروجی می‌برد.

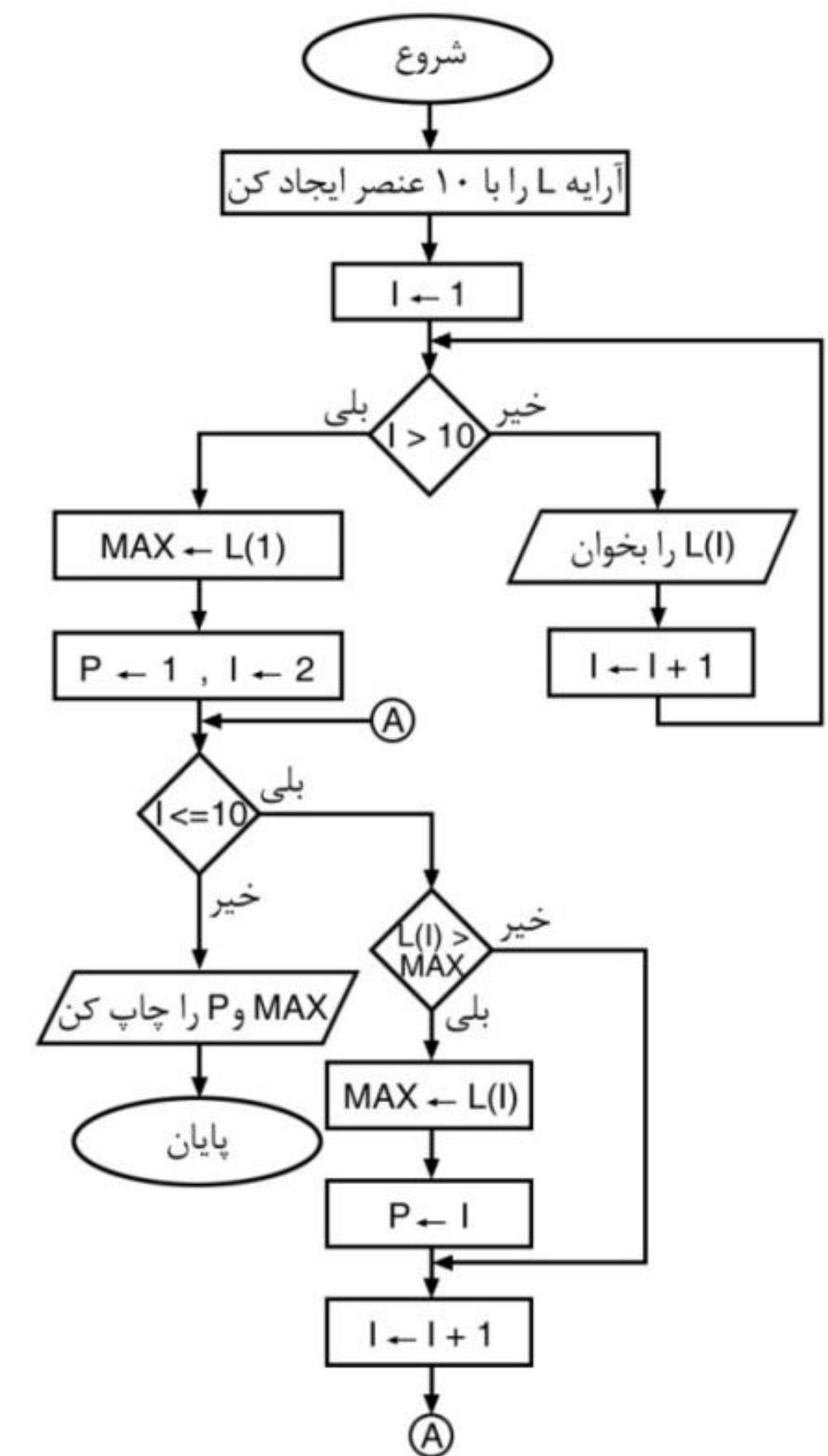


متغیرها	
L	آرایه
I	شمارنده (اندیس آرایه)
X	عدد خوانده شده

الگوریتم

۱. آرایه‌ی L را با ۱۰ عنصر در نظر بگیر
۲. $1 \leftarrow i$
۳. تا زمانی که $i <= 10$ باشد، دستورات ۴ تا ۶ را اجرا کن
۴. X را بخوان
۵. $L(i) \leftarrow X$
۶. $i + 1 \leftarrow i$
۷. پایان حلقه (شروع از مرحله‌ی ۳)
۸. $i \leftarrow 10$
۹. تا زمانی که $i > 1$ است، دستورات ۱۰ تا ۱۱ را اجرا کن
۱۰. $(i)L$ را چاپ کن
۱۱. $i - 1 \leftarrow i$
۱۲. پایان حلقه (شروع از ۹)
۱۳. پایان

فلوچارت



متغیرها

L	آرایه
I	شمارنده
X	بزرگ‌ترین عنصر
P	محل بزرگ‌ترین عنصر

الگوریتمی که ۱۰ عدد را خوانده در آرایه‌ای قرار می‌دهد، سپس بزرگ‌ترین عنصر آرایه را مشخص کرده، تعیین می‌کند که این عنصر در کجا آرایه قرار دارد.

الگوریتم

۱. آرایه‌ی L را با 10 عنصر ایجاد کن
۲. $I \leftarrow 1$
۳. تا زمانی که $10 \leq I$ است، مراحل ۴ تا ۵ را اجرا کن
۴. $L(I)$ را بخوان
۵. $I \leftarrow I + 1$
۶. پایان حلقه (شروع از مرحله ۳)
۷. $MAX \leftarrow L(1)$
۸. $P \leftarrow 1$
۹. $I \leftarrow 2$
۱۰. تا زمانی که $10 \leq I$ است، دستورات ۱۱ تا ۱۲ را اجرا کن
۱۱. اگر $L(I) > MAX$ باشد، آنگاه:

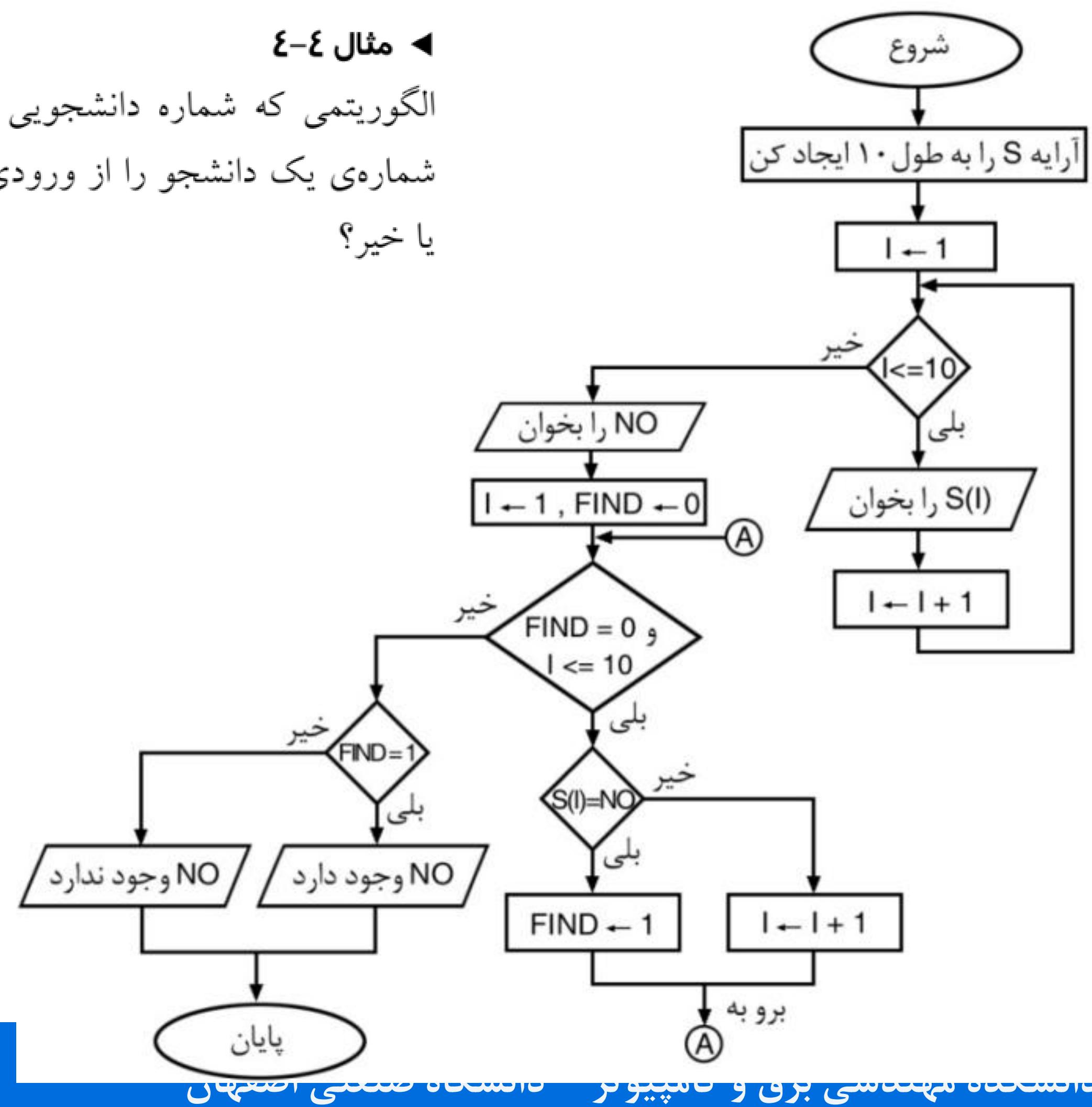
 - $MAX \leftarrow L(I)$
 - $P \leftarrow I$
 - $I \leftarrow I + 1$

۱۳. پایان حلقه $I = 10$ (شروع از دستور ۱۰)
۱۴. MAX و P را چاپ کن
۱۵. پایان

◀ مثال ۴-۴

الگوریتمی که شماره دانشجویی ۱۰ دانشجو را از ورودی خوانده در آرایه‌ای قرار می‌دهد، سپس شماره‌ی یک دانشجو را از ورودی می‌خواند و مشخص می‌کند که آیا این شماره در آرایه وجود دارد

یا خیر؟

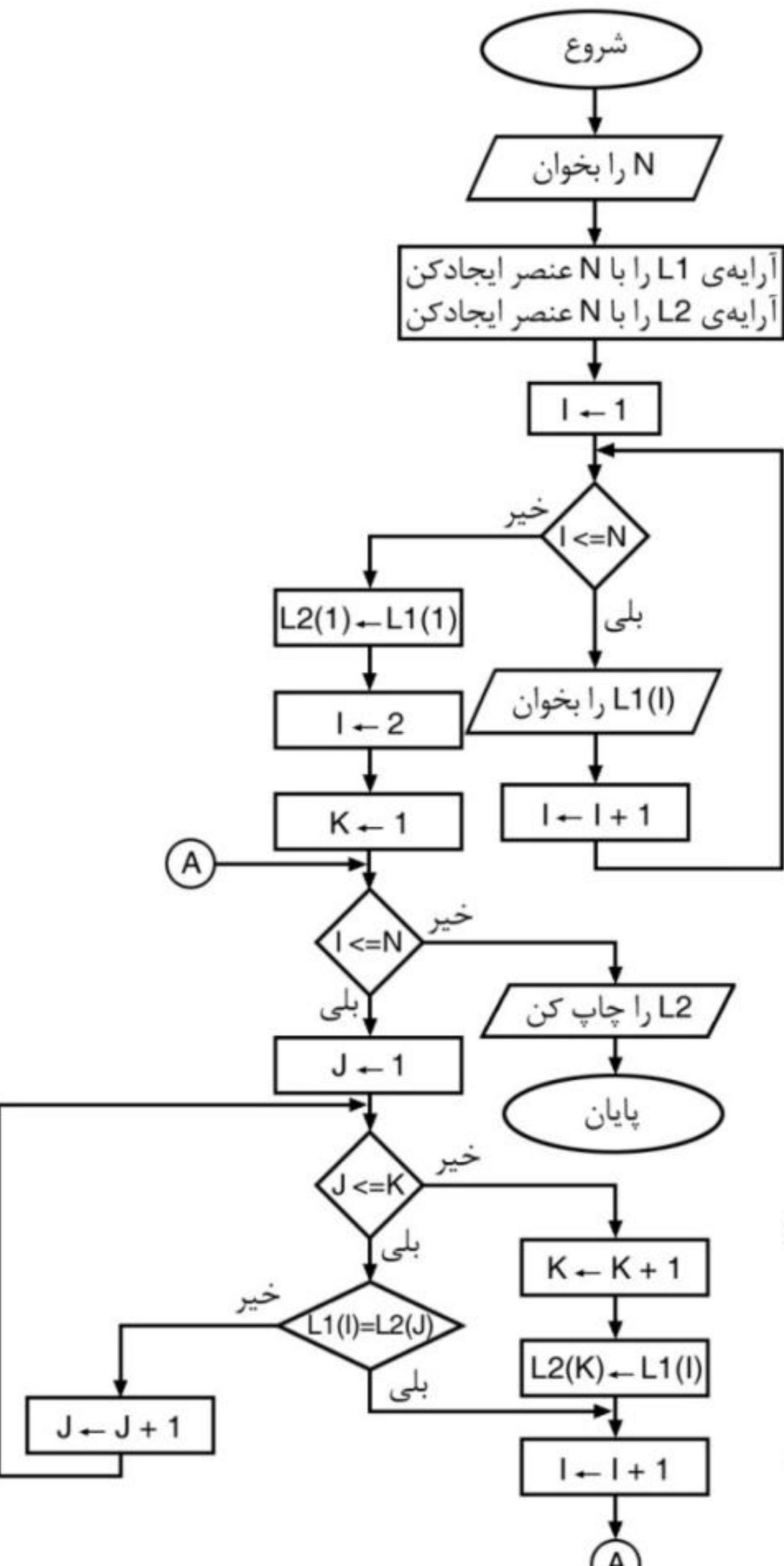


◀ مثال ۴

الگوریتمی که تعدادی عدد را خوانده، در آرایه‌ای قرار می‌دهد. سپس عناصر غیرتکراری آن را در آرایه‌ی دیگری قرار می‌دهد.

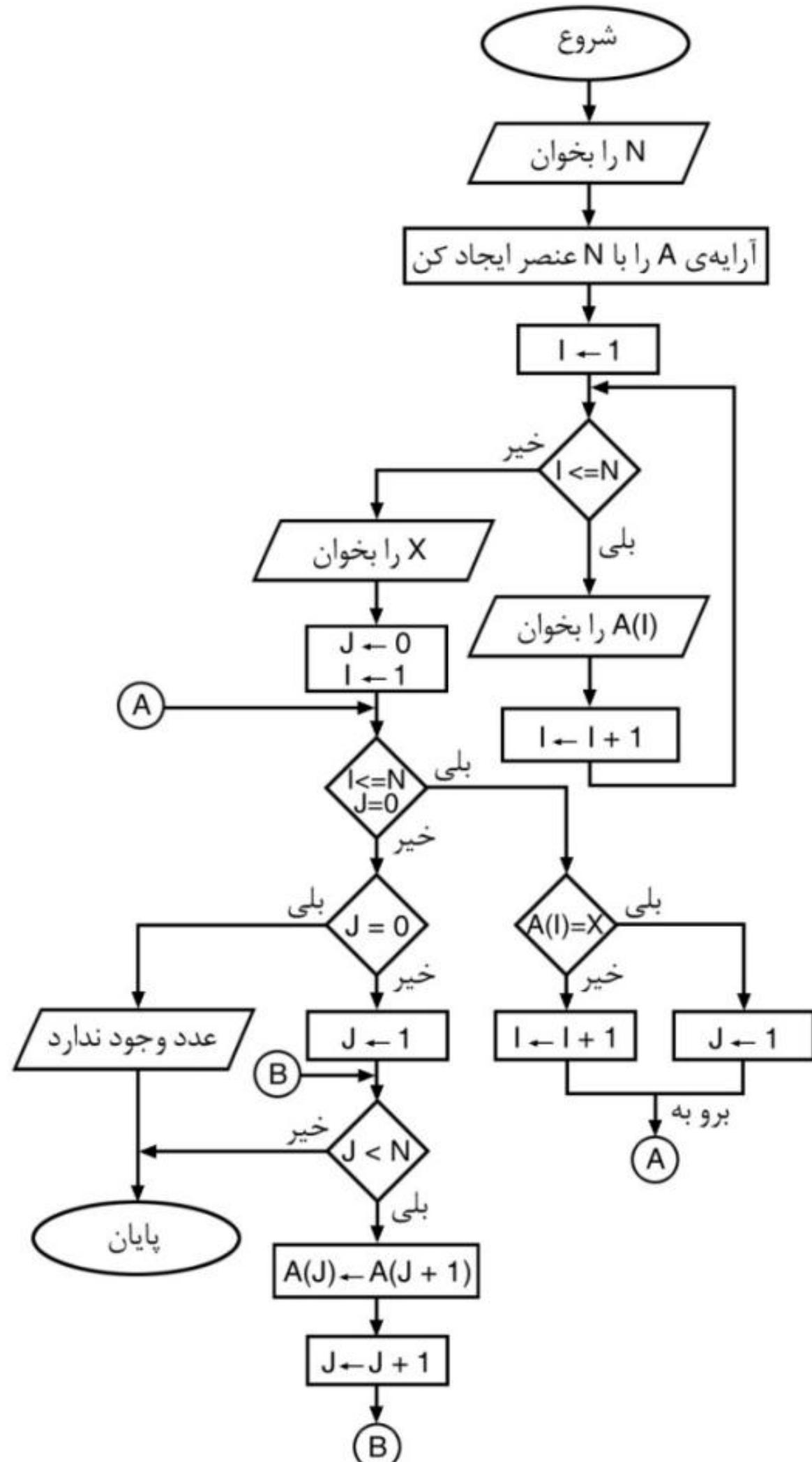
توضیح: برای درک صورت مسئله فرض کنید محتویات آرایه به صورت زیر است:

1	4	4	1	3	2	1	5	9
---	---	---	---	---	---	---	---	---



۸-۴ مثال

الگوریتمی که تعداد N عدد را خوانده در آرایه‌ای قرار می‌دهد. سپس عددی را از ورودی می‌خواند و چنانچه آن عدد در آرایه وجود داشته باشد، آن را حذف می‌کند.



بالای نقطه B دستور زیر جایگزین شود

$$J \leftarrow I$$

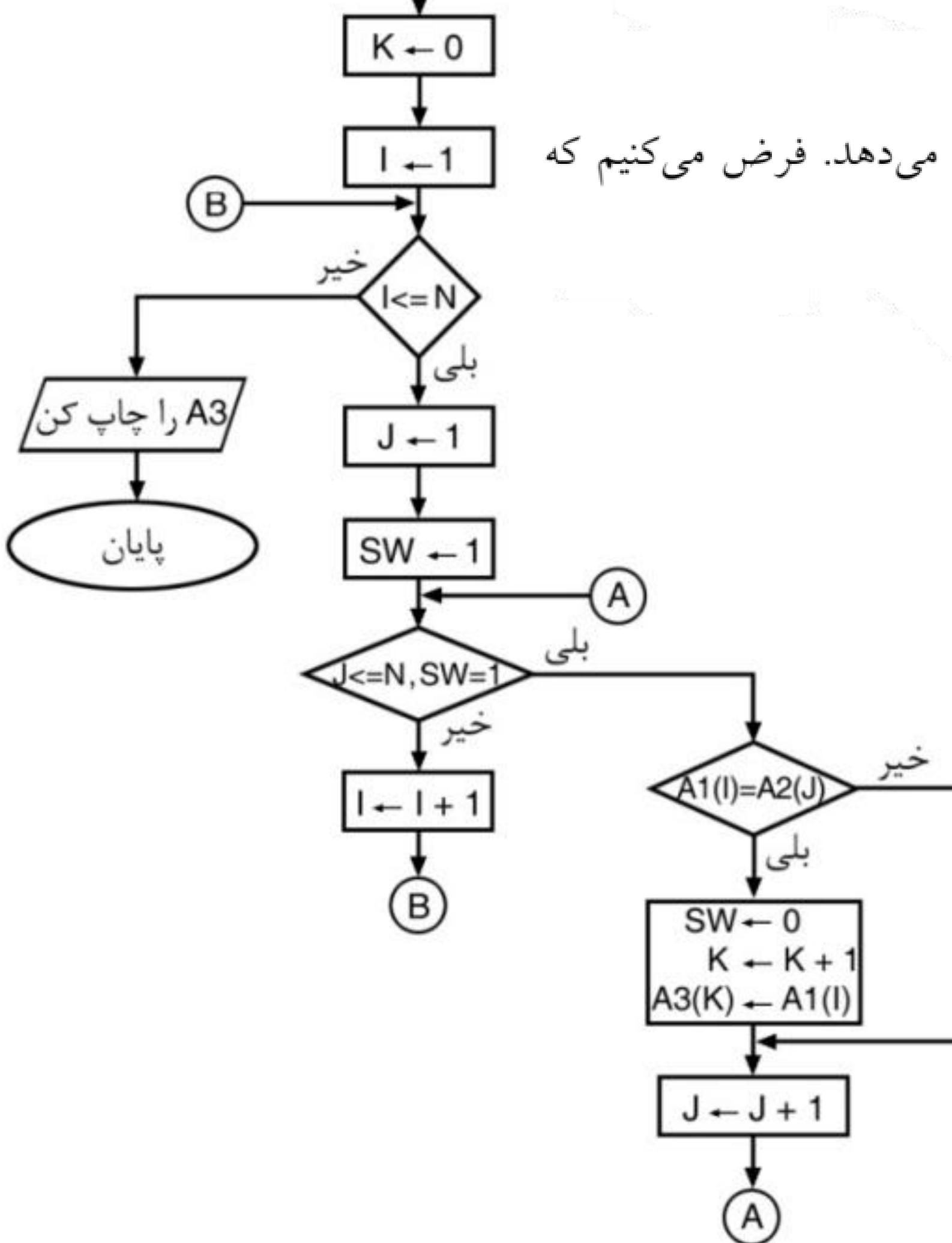
در شرط $N > j$ بخش خیر، دستور زیر اضافه شود

$$N \leftarrow N - 1$$

نکته: این الگوریتم فقط اولین رخداد عدد را حذف می‌کند و رخدادهای تکراری را حذف نمی‌کند

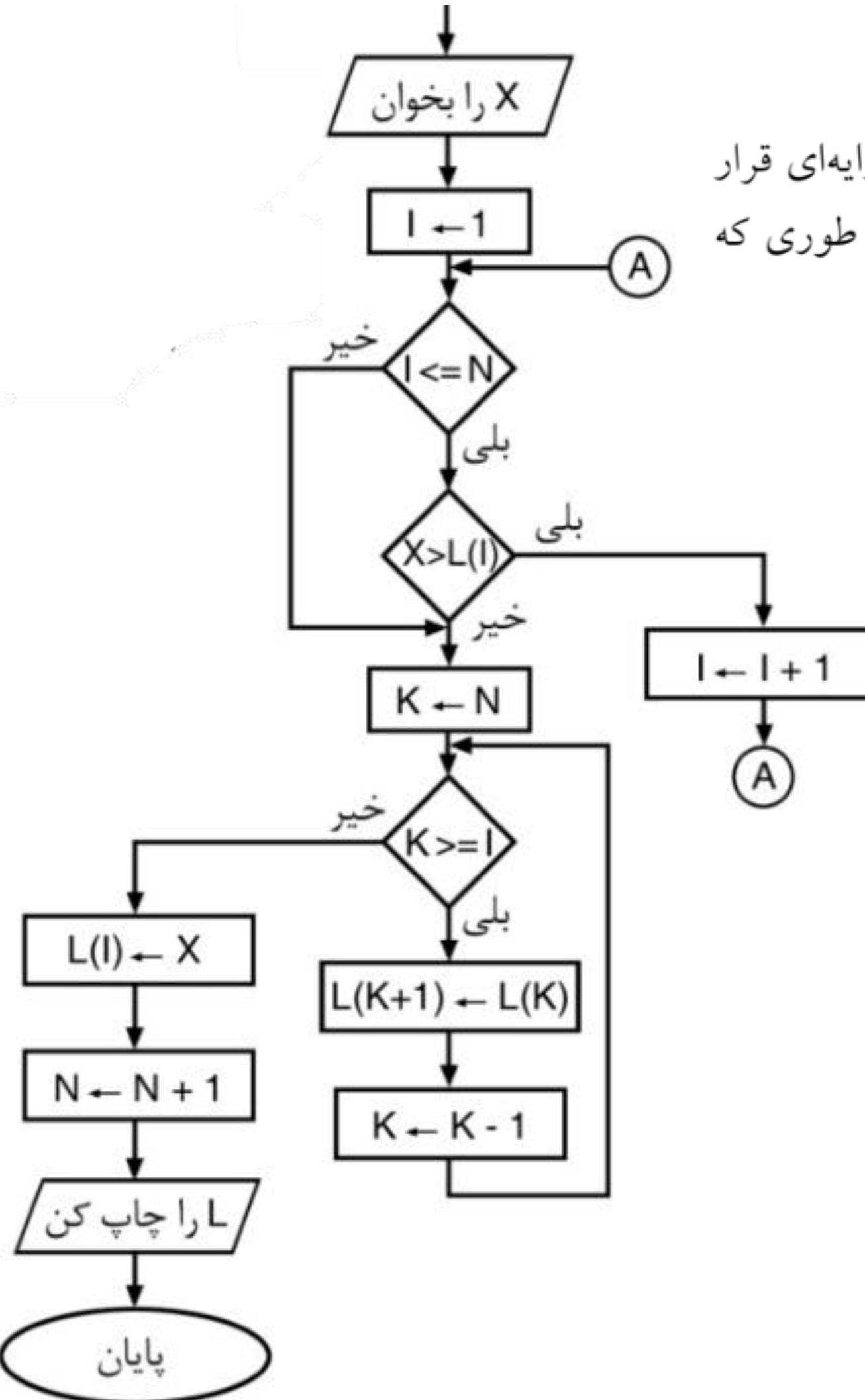
۱۳-۴ مثال ◀

الگوریتمی که عناصر مشترک دو آرایه‌ی A_1 و A_2 را در آرایه‌ی A_3 قرار می‌دهد. فرض می‌کنیم که در هیچ کدام از آرایه‌های A_1 و A_2 عناصر تکراری وجود ندارد.



◀ مثال ۴-۴

الگوریتمی که تعدادی عدد را که به صورت صعودی مرتب هستند از ورودی خوانده، در آرایه‌ای قرار می‌دهد. سپس یک عدد را از ورودی می‌خواند و در جای مناسبی از آرایه قرار می‌دهد، به طوری که ترتیب صعودی آرایه حفظ شود (حداکثر تعداد اعداد ۲۰۰ ولی تعداد واقعی آنها N است).



◀ مثال ۴

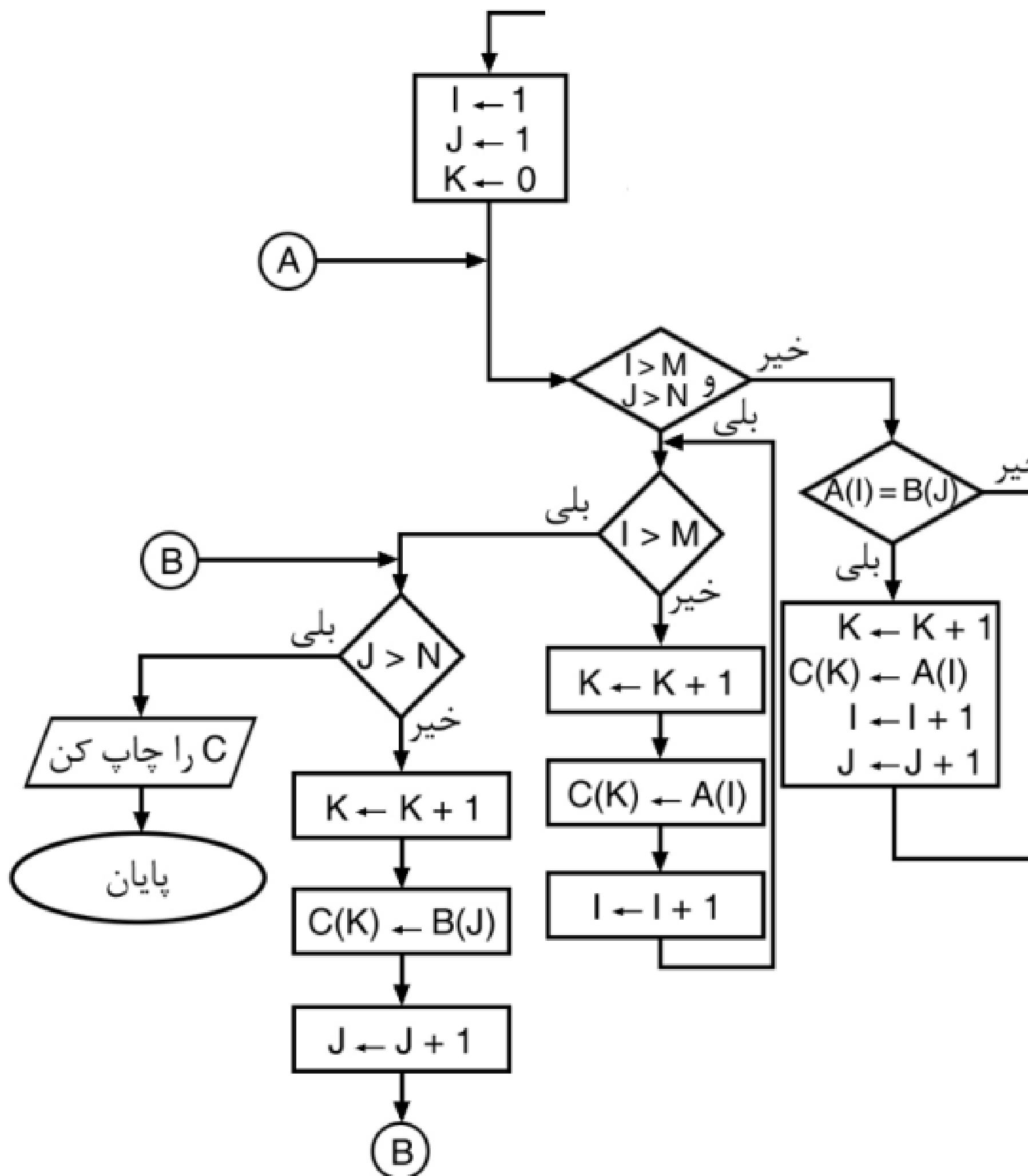
الگوریتمی که دو لیست مرتب A و B را که تعداد عناصر آنها به ترتیب M و N هستند خوانده و در لیست C در هم ادغام می‌کند (لیست حاصل باید مرتب باشد). فرض می‌کنیم که تعداد عناصر حداکثر برابر با ۲۰۰ است). برای درک صورت مسئله فرض کنید لیست‌های A و B به صورت زیر باشند.

A	4	9	10	12	17	22
---	---	---	----	----	----	----

B	5	9	11	14	19	20
---	---	---	----	----	----	----

پس از ادغام دو لیست، لیست C به صورت زیر تشکیل می‌شود:

C	4	5	9	10	11	12	14	17	19	20	22
---	---	---	---	----	----	----	----	----	----	----	----



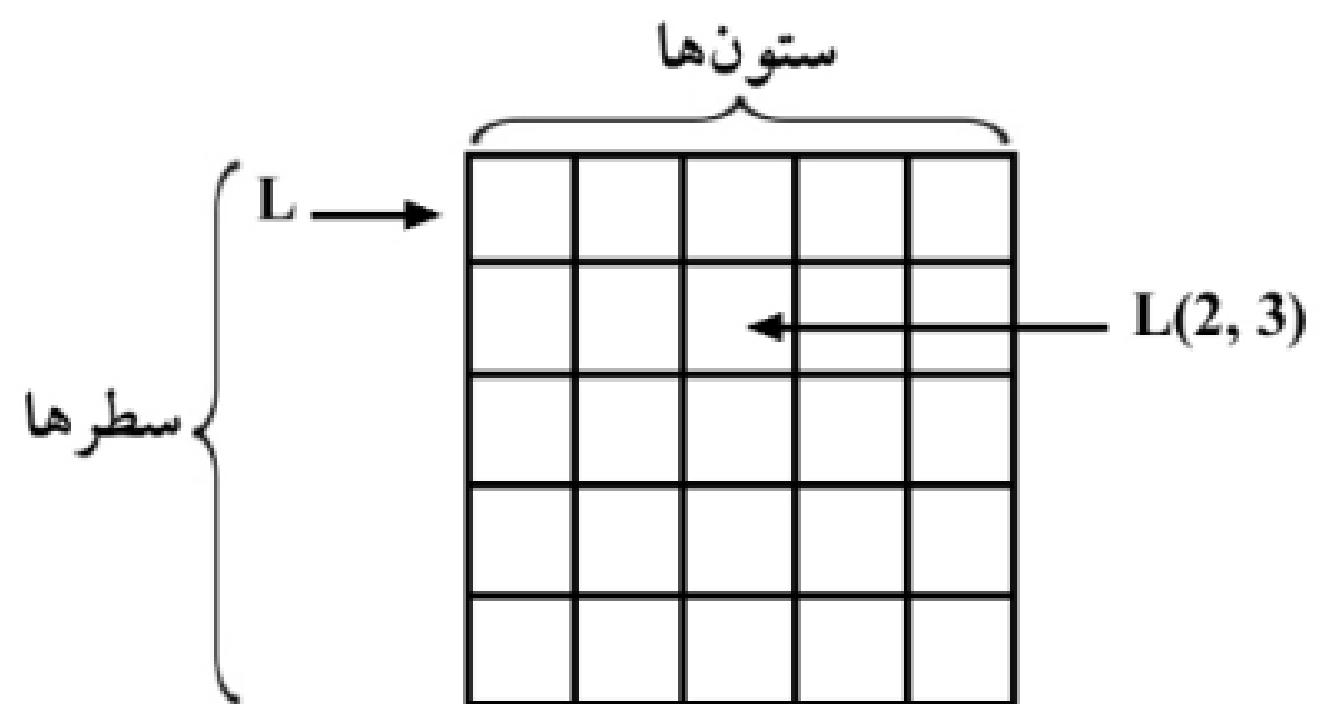
A	4	9	10	12	17	22
---	---	---	----	----	----	----

B	5	9	11	14	19	20
---	---	---	----	----	----	----

پس از ادغام دو لیست، لیست C به صورت زیر تشکیل می‌شود:

C	4	5	9	10	11	12	14	17	19	20	22
---	---	---	---	----	----	----	----	----	----	----	----

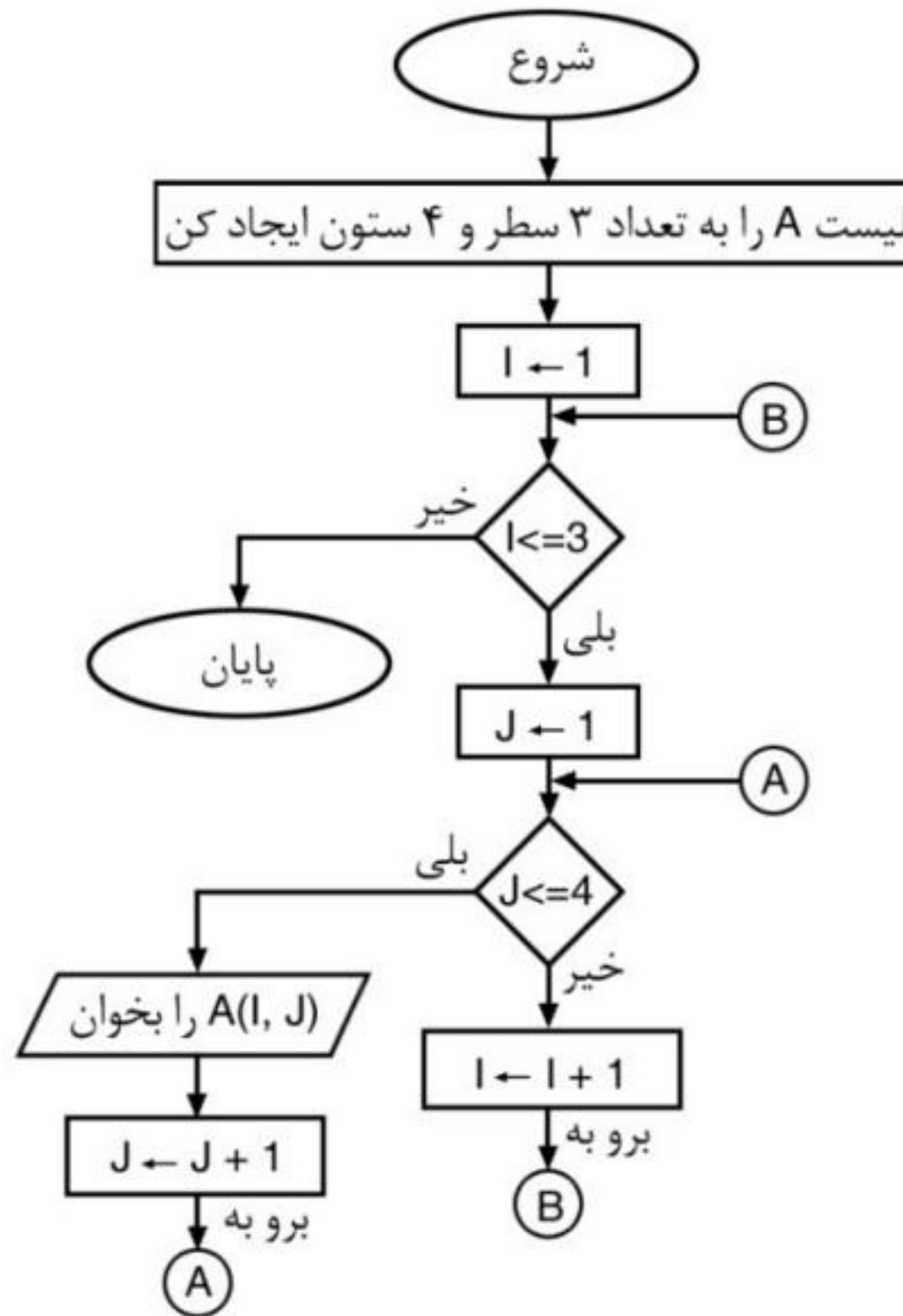
آرایه های چند بعدی



شکل ۴-۴ ◀ نمونه‌ای از آرایه‌ی دو بعدی

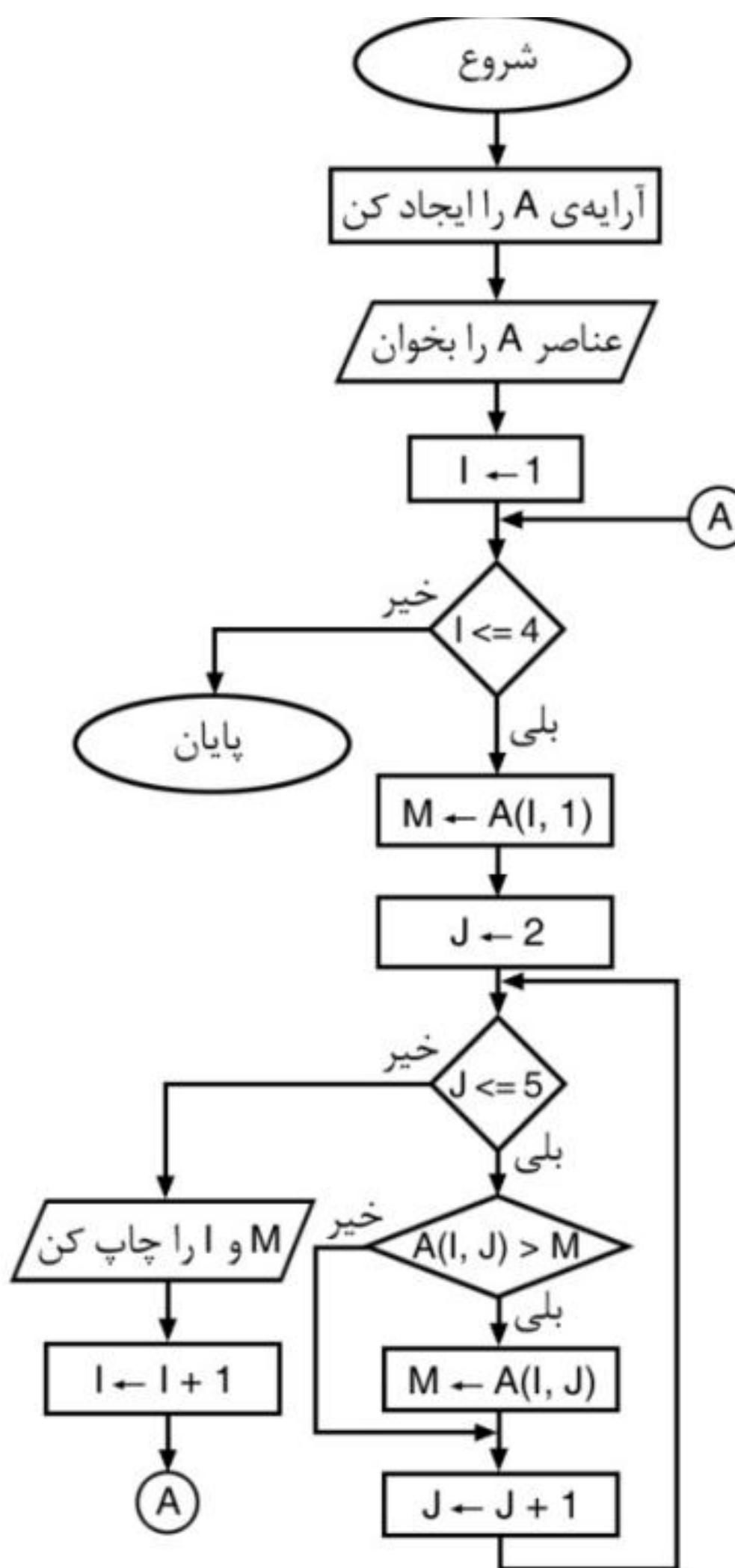
۱۸-۴ ◀

الگوریتم خواندن عناصر آرایه‌ی دو بعدی به نام A.



◀ مثال ۴

الگوریتمی که عناصر آرایه‌ی 5×4 را خوانده، بزرگ‌ترین عنصر هر سطر را پیدا کرده، به همراه شماره‌ی سطر در خروجی چاپ می‌کند.

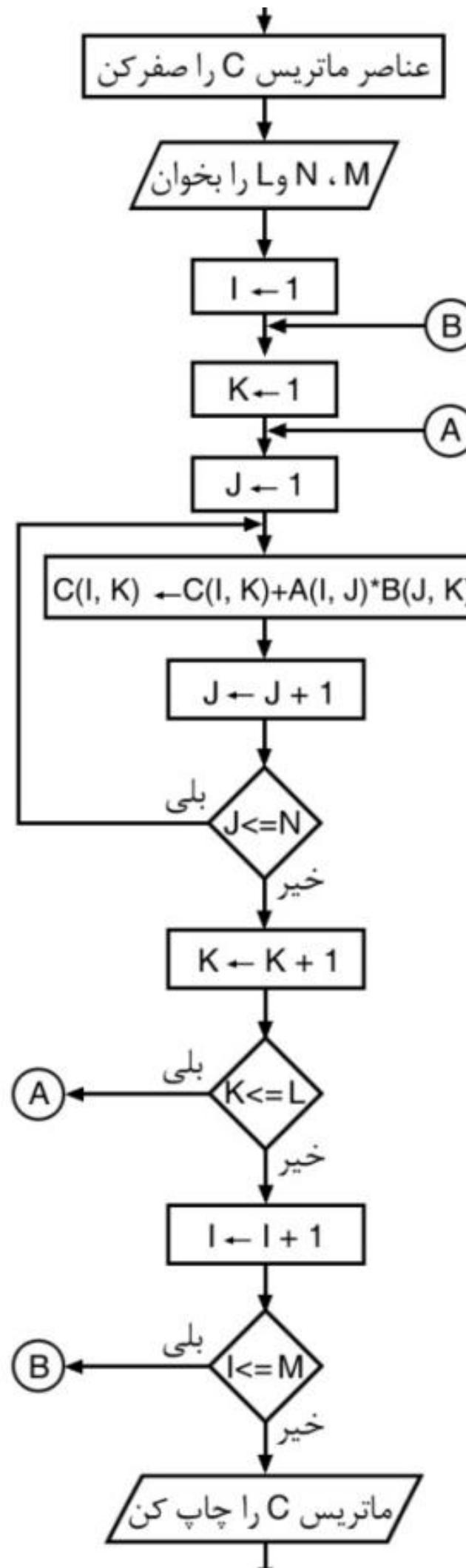


الگوریتمی که حاصل ضرب دو ماتریس A و B را محاسبه می‌کند و در ماتریس C قرار می‌دهد. ماتریس A دارای M سطر و N ستون ماتریس B دارای N سطر و L ستون است. ماتریس A حداقل ۱۰ سطر و ۲۰ ستون و ماتریس B دارای ۲۰ سطر و ۲۰ ستون است. برای درک روش ضرب دو ماتریس مثالی در نظر گرفته می‌شود.

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 2 & 3 \end{bmatrix}$$

$$B = \begin{bmatrix} 3 & 2 & 5 \\ 4 & 1 & 7 \end{bmatrix}$$

$$C = A \times B = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 2 & 3 \end{bmatrix} \times \begin{bmatrix} 3 & 2 & 5 \\ 4 & 1 & 7 \end{bmatrix} = \begin{bmatrix} 15 & 5 & 26 \\ 22 & 8 & 38 \\ 18 & 7 & 31 \end{bmatrix}$$



مبانی کامپیووتر و الگوریتم‌ها

دکتر عین‌الله جعفر نژاد قمی

(عضو هیأت علمی جهاد دانشگاهی)

دکتر امیر کریم‌پور

(عضو پژوهشی دانشگاه کنرادو)



مرجع