مبانی کامپیوتر و برنامه‌سازی

Algorithm-part 2 of 2

میثم سعیدی-استادیار گروه مهندسی مکانیک

Razi University
Educational Use

دستورالعمل‌های شرطی

در الگوریتم‌هایی که مشاهده کردید، دستورالعمل‌های الگوریتم، به ترتیب، از اولین دستور به آخرین دستور اجرا شدند. اما الگوریتم‌هایی وجود دارند که در آنها، براساس شرایطی که اتفاق می‌افتد، دستورات خاصی اجرا می‌شوند و از اجرای بعضی از آنها صرفنظر می‌گردد. برای پیاده‌سازی این گونه شرایط، از دستورالعمل‌های شرطی استفاده می‌شود. دستورالعمل‌های شرطی با کلمه **اگر** شروع می‌شوند. نمونه‌ای از دستور شرطی، در زیر مشاهده می‌گردد:

اگر شرط **آنگاه** دستورات

Razi University
Educational Use

در این دستور شرطی، اگر **شرطِ** ذکر شده ارزش درستی داشته باشد، دستوراتی که در جلوی آن ذکر شده‌اند اجرا می‌گردد.

نمونه دیگری از دستورالعمل‌های شرطی به صورت زیر است:

اگر شرط آنگاه دستورات ۱ وگرنه دستورات ۲

در این نوع دستور شرطی، چنانچه شرط ذکر شده دارای ارزش درستی باشد، **دستورات ۱** و در صورتی که شرط دارای ارزش نادرستی باشد، **دستورات ۲** اجرا می‌شوند.

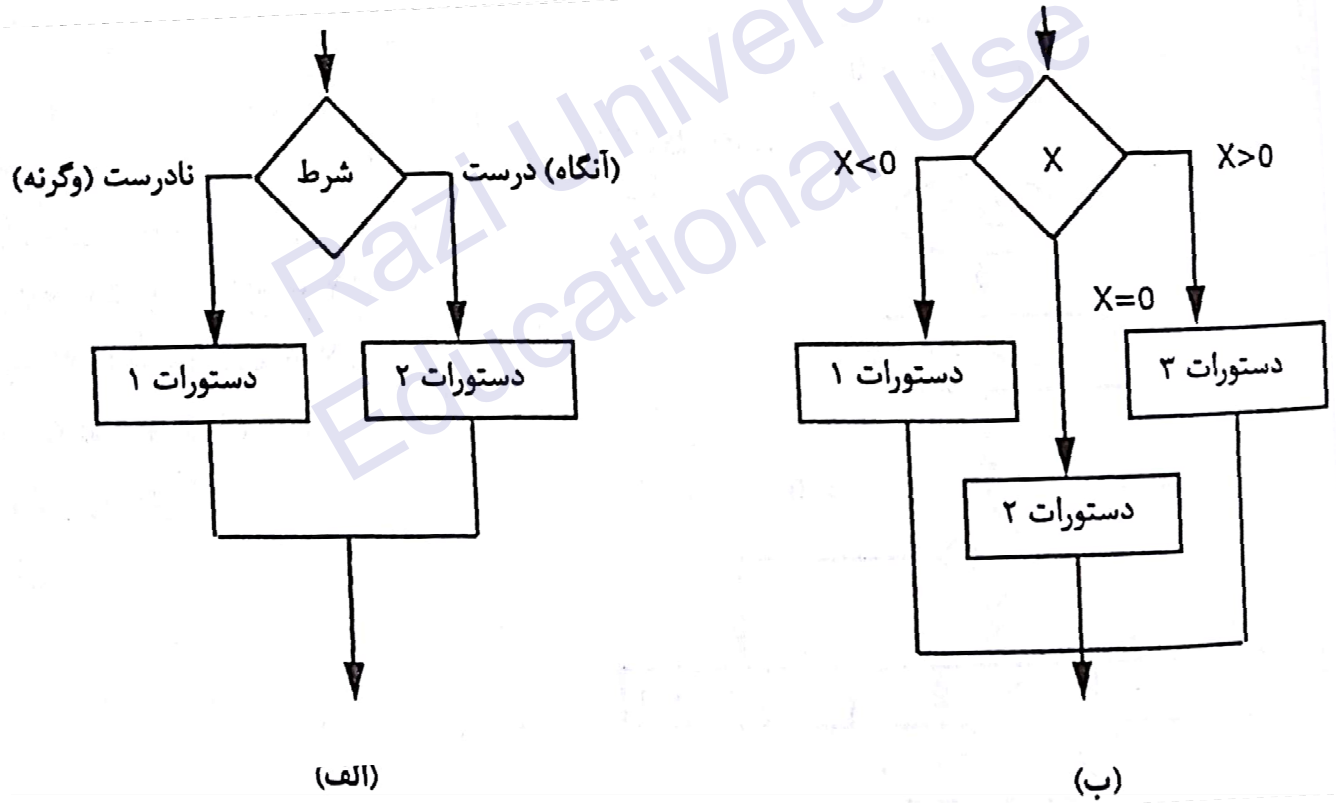
Razi University Educational Use

با توجه به دستورالعمل‌های شرطی، می‌توان الگوریتم را در جایی خاتمه داده، از جای دیگری ادامه داد. بدین‌ترتیب، می‌توان کنترل اجرای الگوریتم را از نقطه‌ای به نقطهٔ دیگر منتقل کرد، به طوری که اجرای الگوریتم از نقطهٔ جدید ادامه یابد. این کار را **انتقال کنترل** گویند. انتقال کنترل می‌تواند شرطی یا غیرشرطی باشد در انتقال کنترل شرطی، کنترل الگوریتم تحت شرایط خاصی از جایی به جای دیگر منتقل می‌شود. مثل **اگر A > B بود، برو به مرحله ۵.** اما در انتقال کنترل غیرشرطی، کنترل اجرای الگوریتم، در هر حال، به نقطهٔ دیگری می‌رود. مثل دستورِ **برو به مرحله ۱۰**، که کنترل الگوریتم را بدون هیچ شرطی به مرحله ۱۰ می‌برد.

Razi University Educational Use

علامت های شرطی در فلوچارت

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



(الف) (ب)

Razi University Educational Use

▪ مثال ۱۰

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

متغیرها

مقدار اول	X
مقدار دوم	Y
مقدار بزرگتر	MAX

الگوریتم

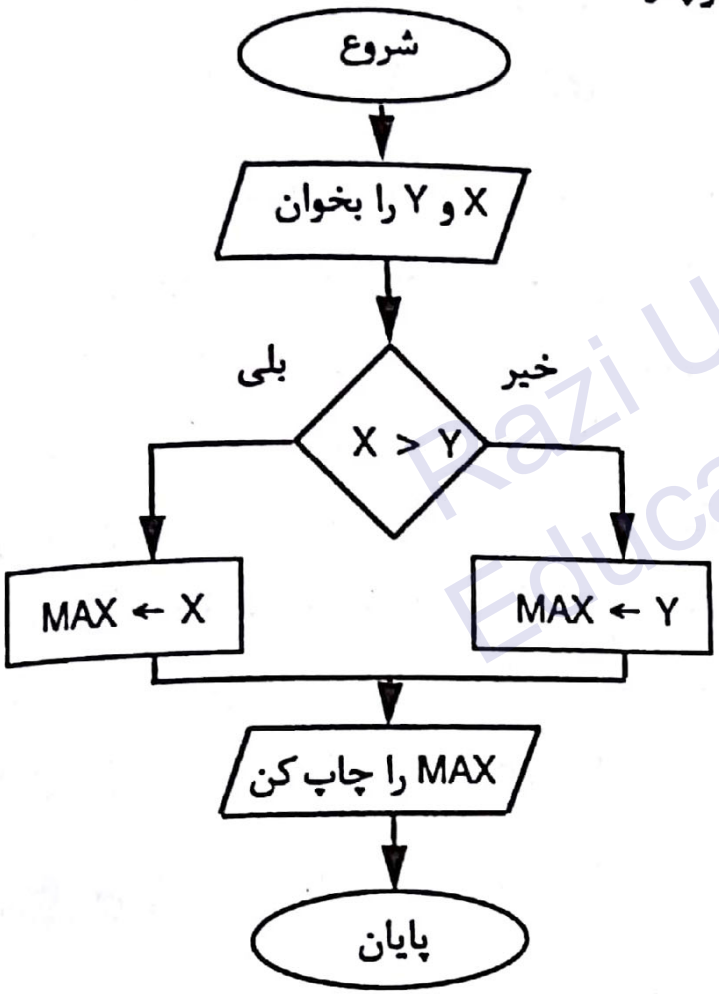
۱. X و Y را بخوان

۲. اگر X>Y آنگاه MAX ← X وگرنه MAX ← Y

۳. MAX را چاپ کن

۴. پایان

فلوچارت



Razi University
Educational Use

▪ مثال ۱۱

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

$X < 0$	$Y = 3X - 5$
$X = 0$	$Y = 2$
$X > 0$	$Y = 2X + 1$

الگوریتم

۱. X را بخوان

۲. اگر X<0 آنگاه Y ← 3 * X - 5

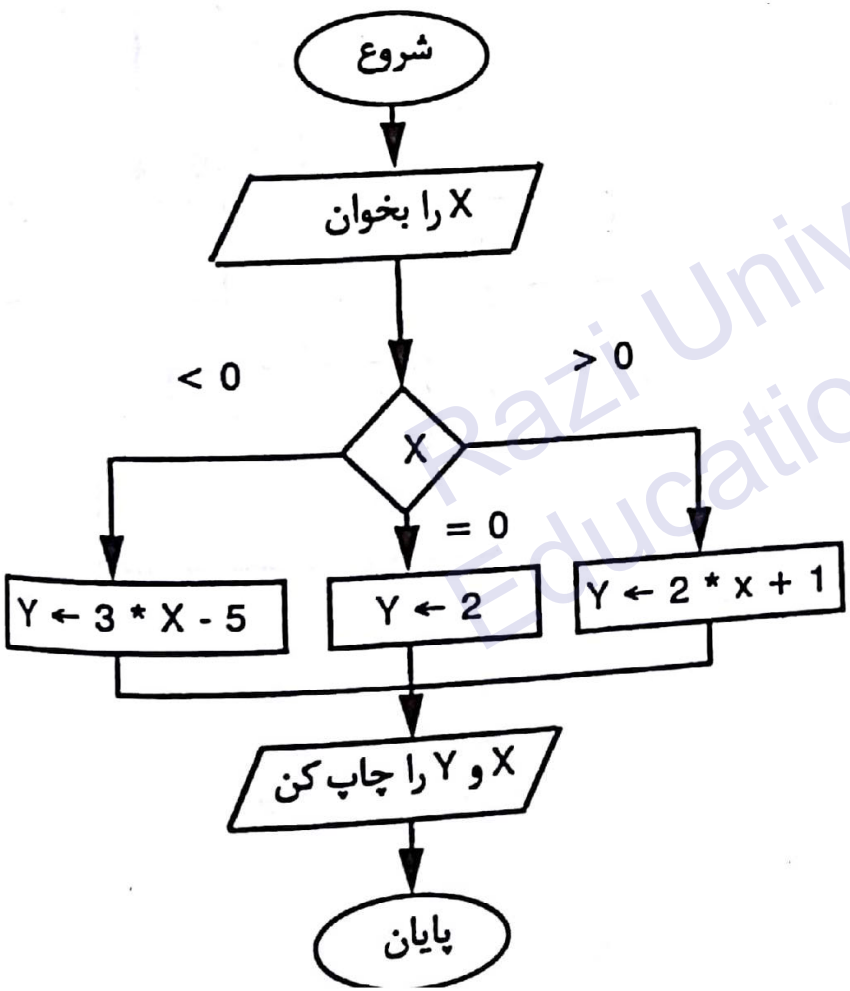
وگرنه، اگر X = 0 آنگاه Y ← 2

وگرنه Y ← 2 * X + 1

۳. X و Y را چاپ کن

۴. پایان

فلوچارت



Razi University
Educational Use

▪ مثال ۱۲

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.

توضیح

برای اینکه تشخیص دهیم عدد X زوج است یا خیر، آن را بر ۲ تقسیم می‌کنیم. اگر باقیمانده تقسیم صحیح بر ۲، برابر با صفر باشد، عدد زوج است وگرنه فرد است. برای به دست آوردن باقیمانده تقسیم، خارج قسمت را در ۲ (مقسوم علیه) ضرب کرده، از X (مقسوم) کم می‌کنیم.

Razi University
Educational Use

متغیرها

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

الگوریتم

۱. X را بخوان

۲. D ← [X / 2]

۳. R ← X - D * 2

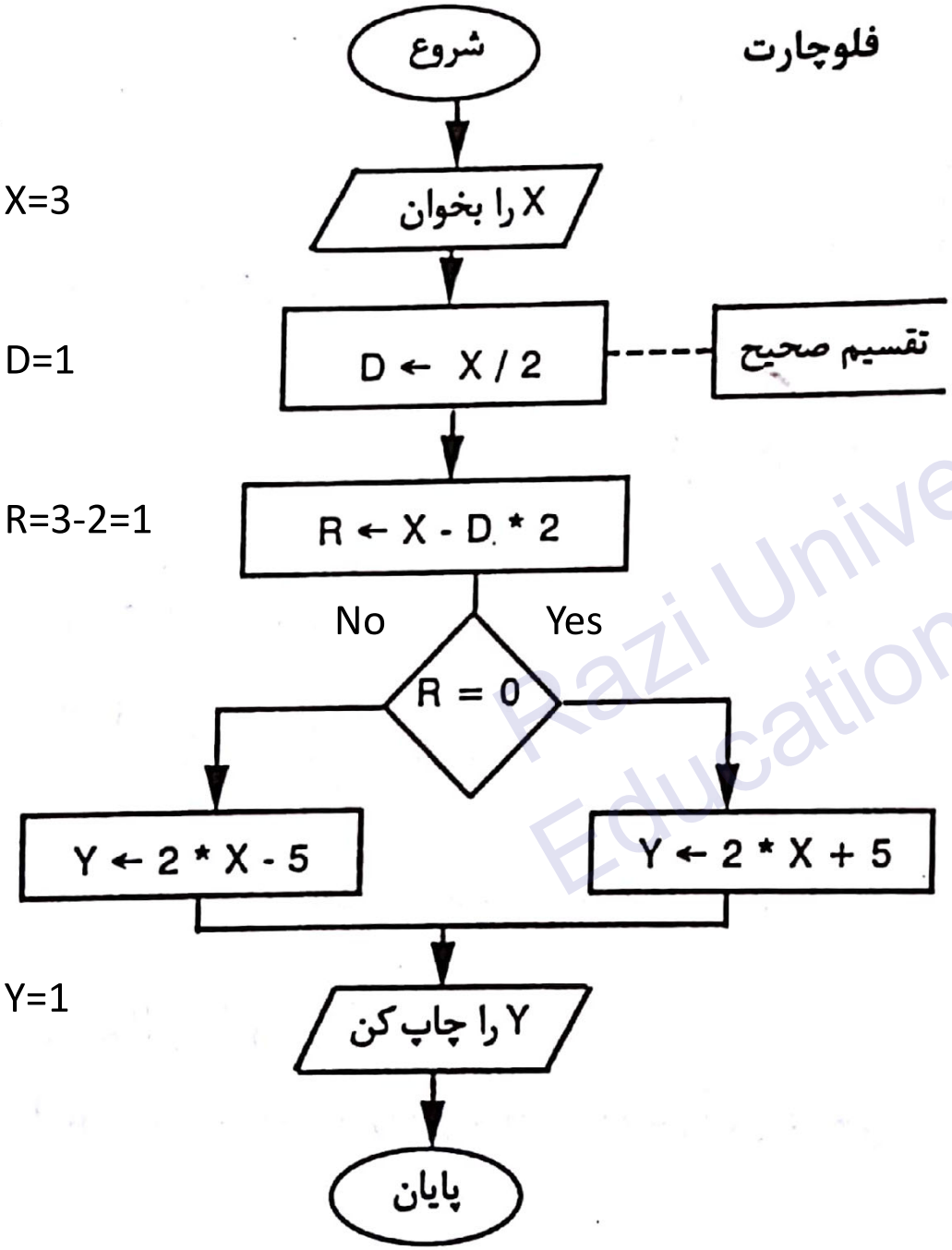
۴. اگر R = 0 آنگاه Y ← 2 * X + 5

وگرنه، Y ← 2 * X - 5

۵. Y را چاپ کن

۶. پایان

فلوچارت



Razi University
Educational Use

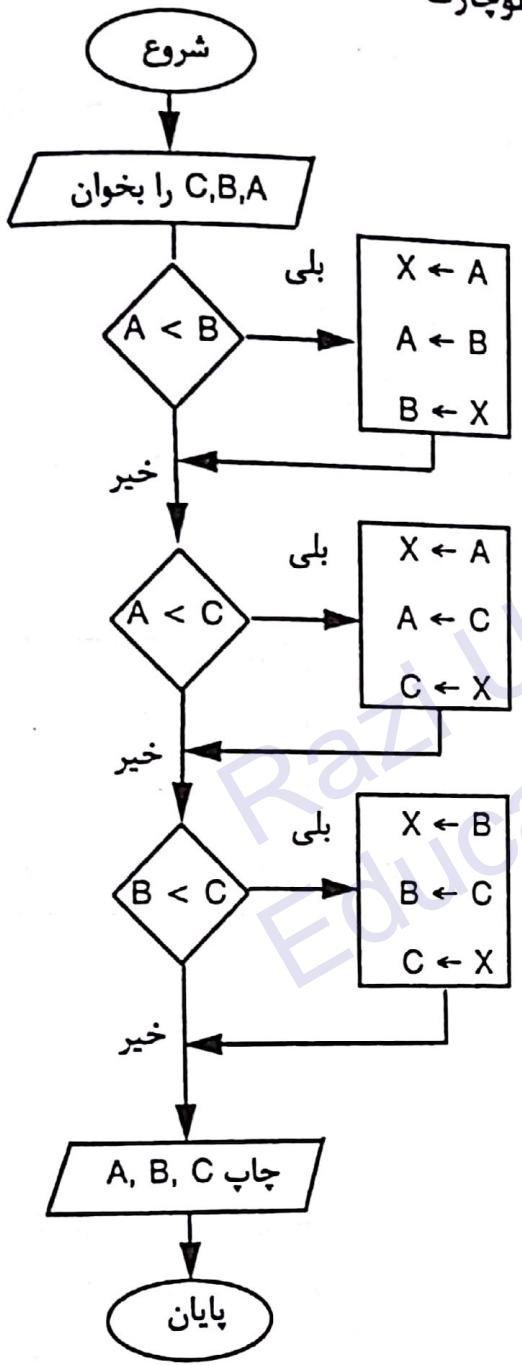
- **مثال ۱۳**

الگوریتمی که ۳ مقدار را از ورودی خوانده، آنها را به ترتیب نزولی در خروجی چاپ می‌کند. منظور از ترتیب نزولی این است که اعداد از بزرگ به کوچک در خروجی چاپ شوند.

Razi University
Educational Use

A=1 AA=2
B=2 BB=1
C=3 CC=3

فلوچارت



A=2
B=1
C=3

A=3 AA=3
B=1 BB=1
C=2 CC=2

A=3 AA=3
B=2 BB=2
C=1 CC=1

متغیرها	
اعداد خوانده شده	A,B,C
متغیر کمکی	X

الگوریتم

۱. C,B,A را بخوان
۲. اگر A < B آنگاه
X ← A
A ← B
B ← X
۳. اگر A < C آنگاه
X ← A
A ← C
C ← X
۴. اگر B < C آنگاه
X ← B
B ← C
C ← X
۵. A, B, C را چاپ کن
۶. پایان

Raz University Educational Use

▪ مثال ۱۴

الگوریتمی که عددی را خوانده، قدرمطلق آن را در خروجی چاپ می‌کند.

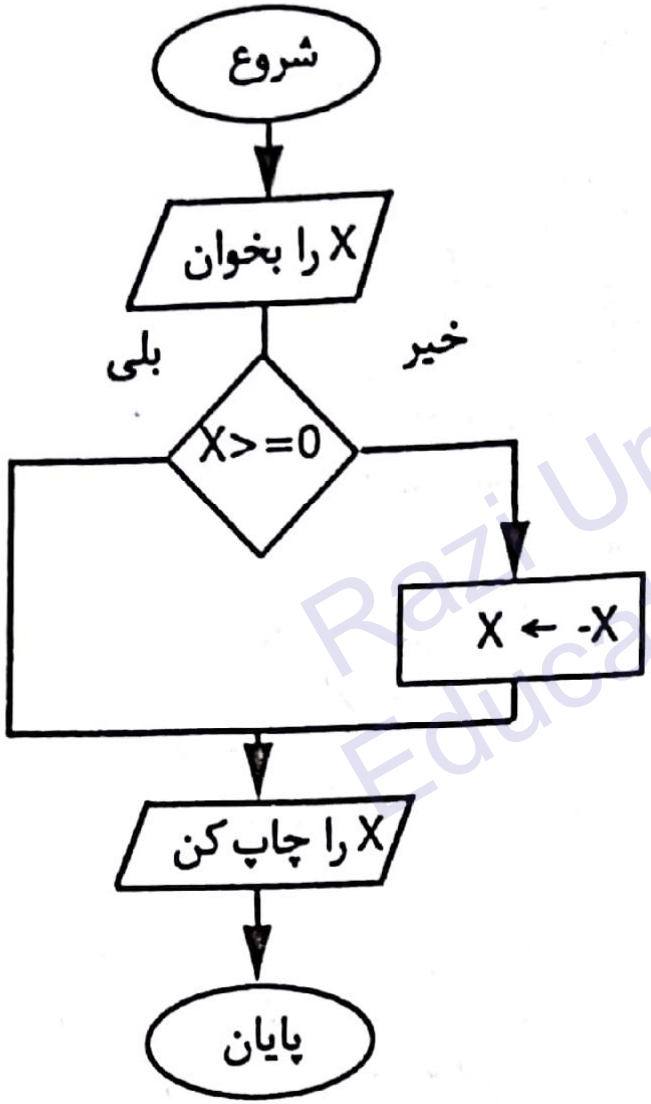
توضیح

اگر عدد مثبت باشد قدرمطلق آن عدد با خودش برابر است ولی اگر عدد منفی باشد، آن را در یک منفی ضرب می‌کنیم تا به مثبت تبدیل شود. عدد را X نامگذاری می‌کنیم.

در این مثال، یک دستور انتقال کنترل شرطی وجود دارد. در دستور شماره ۲، اگر X >= 0 باشد آنگاه کنترل الگوریتم به مرحله ۴ می‌رود تا X را چاپ کرده، در مرحله ۵ متوقف شود. اما اگر شرط X >= 0 ارزش درستی نداشته باشد (عدد منفی باشد)، مرحله ۳ اجرا می‌شود تا عدد در یک منفی ضرب شود و به مثبت تبدیل گردد. از آنجا مرحله ۳ به بعد اجرا می‌شود.

Razi University Educational Use

فلوچارت



الگوریتم

۱. X را بخوان
۲. اگر X > 0 برو به مرحله ۴
۳. X ← -X
۴. X را چاپ کن
۵. پایان

Razi University
Educational Use

▪ مثال ۱۵

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

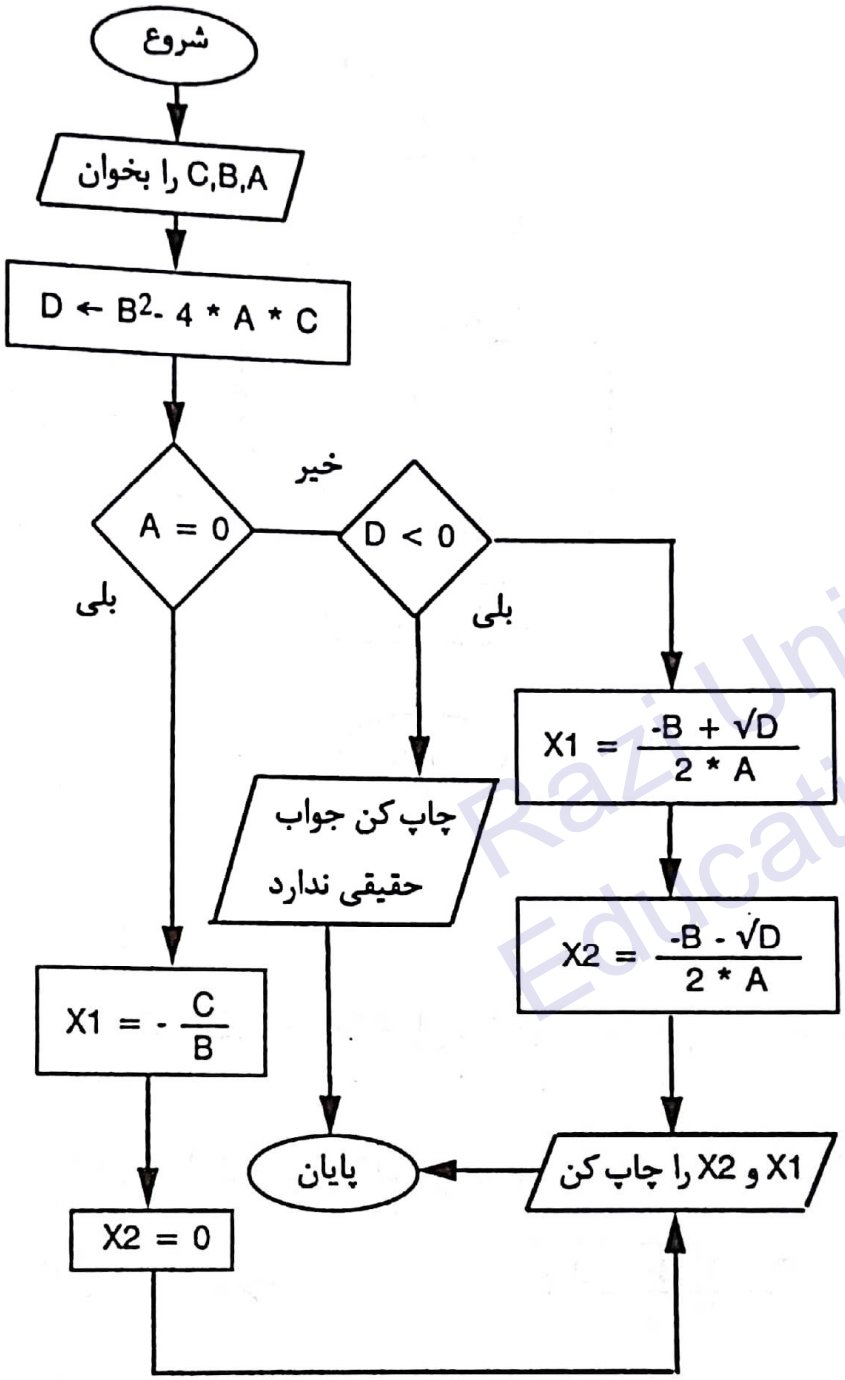
توضیح

برای محاسبه جواب‌های حقیقی این نوع معادلات، از رابطه زیر استفاده می‌شود:

$$X1 = \frac{-B + \sqrt{B^2 - 4AC}}{2 * A} \qquad X2 = \frac{-B - \sqrt{B^2 - 4AC}}{2 * A}$$

بدیهی است که اگر حاصل عبارت $B^2 - 4AC$ (داخل رادیکال) منفی باشد، معادله جوابهای حقیقی ندارد.

Razi University Education Not Use



الگوریتم

۱. C,B,A را بخوان

۲. $D \leftarrow B^2 - 4 * A * C$

۳. اگر $A = 0$ آنگاه $X1 = -\frac{C}{B}$ و $X2 = 0$ برو به ۷

۴. اگر $D < 0$ آنگاه چاپ کن "جواب حقیقی ندارد" و برو به ۸

۵. $X1 = \frac{-B + \sqrt{D}}{2 * A}$

۶. $X2 = \frac{-B - \sqrt{D}}{2 * A}$

۷. X1 و X2 را چاپ کن

۸. پایان

Raz University Educational Use

اثبات درستی الگوریتم

یکی از مراحل مهم در طراحی الگوریتم، اثبات درستی الگوریتم است. به عبارت دیگر باید الگوریتم را به روش دستی و با داده‌های آزمایشی امتحان کرد تا مشخص شود که آیا الگوریتم به ازای آن داده‌های آزمایشی درست عمل کرد یا خیر. اگر درست عمل کرد، تا حدود زیادی مطمئن هستیم که الگوریتم به درستی عمل می‌کند، ولی اگر به ازای داده‌های آزمایشی به درستی عمل نکرد، باید اشکالات موجود را پیدا کرده، آنها را برطرف نمود و سپس الگوریتم را امتحان کرد. برای امتحان الگوریتم می‌توان فلوچارت زیر را در نظر گرفت.

Razi University
Educational Use

امتحان الگوریتم برای X = 7

شماره دستور	X	D	R	Y	خروجی
1	7				
2		3			
3			1		
4				9	
5					9
6					

با این آزمایش، یک مسیر از الگوریتم امتحان شد و چون مقدار Y با مقدار مورد انتظار، برابر است، نتیجه می‌گیریم این مسیر از الگوریتم به درستی عمل می‌کند. اکنون الگوریتم را به ازای X = 7 که یک مقدار فرد است، امتحان می‌کنیم. انتظار داریم که حاصل عبارت 2 * X - 5 که برابر با 9 = 5 - 7 * 2 است در Y قرار گیرد.

با ورود عدد ۷، مسیر دیگری از الگوریتم اجرا شد و نتیجه‌ای را که در Y قرار گرفت با نتیجهٔ مورد انتظار ما که عدد ۹ است، برابر است. بنابراین، نتیجه می‌گیریم که الگوریتم به درستی عمل می‌کند. به عنوان تمرین، الگوریتم مثال ۱۵ را امتحان کنید.

Razi University Educational Use

حلقه های تکرار

الگوریتم‌هایی که تاکنون نوشته شده‌اند، طوری طراحی شده‌اند که نیاز به تکرار اجـرای دسـتورات نـبوده است. ولی در اغلب الگوریتم‌هایی که کامپیوتر با آنها سروکار دارد، دستوراتی وجود دارند که اجرای آنـها باید چندین بار تکرار شود. به عنوان یک مثال ساده، این مسئله را در نظر بگیرید: دانشـجویی مـی‌خواهـد الگوریتمی بنویسد که تعداد ۱۰۰ عدد را از ورودی خوانده، آنها را با هم جمع کند. فکر می‌کنید که از چه راه حلی استفاده می‌کند؟ یک راه حل این است که ۱۰۰ دستور خواندن در الگوریتم تدارک ببیند و در هر دستور، یک عدد را از ورودی بخواند و آنها را با هم جمع کند و به خروجی ببرد. راه حـل دیگـر ایـن است کـه بـه کامپیوتر بگوید که عمل خواندن عدد و عمل جمع‌کردن (دو دستور به جای ۱۰۰ دستور) را ۱۰۰ بار تکرار کند. با مقایسهٔ سادهٔ این دو روش، نتیجه می‌گیریم کـه روش دوم مـعقول‌تر است. راهکـار تکـرار اجـرای دستورات را **حلقه تکرار** گویند.

Razi University
Educational Use

در حلقه تکرار باید سه چیز مشخص باشد :

۱. **شرط حلقه تکرار** که مشخص می‌کند، حلقه تکرار کِی باید خاتمه یابد. در مثال موردنظر ما، شرط حلقه تکرار این است که ۱۰۰ عدد خوانده شود.

۲. **شمارنده حلقه تکرار**، که مشخص می‌کند تاکنون حلقه چند بار اجرا شده است.

۳. **گام افزایش به حلقه تکرار**، که مقداری است که به ازای هر بار اجرای حلقه، به شمارنده حلقه اضافه می‌شود.

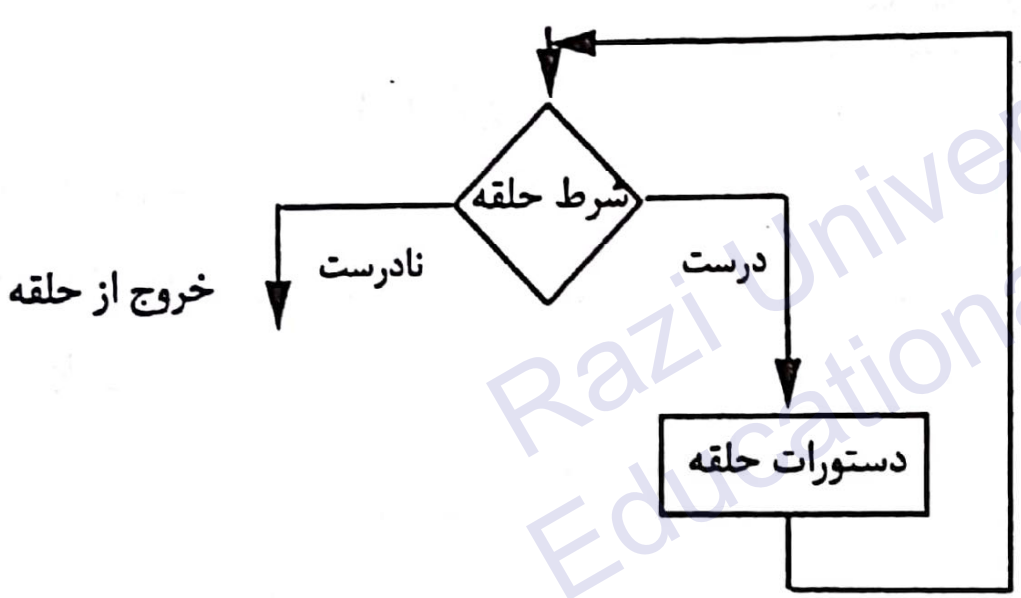
Razi University Educational Use

همانطور که گفته شد، شمارنده‌ها متغیرهایی هستند که تعداد دفعات تکرار دستورات داخل حلقه تکرار را مشخص می‌کنند. شاید تاکنون کنتورهای پمپ بنزین را دیده باشید. شخصی که وارد پمپ بنزین می‌شود تا اتومبیل خود را بنزین بزند، ابتدا کنتور دستگاه را صفر می‌کند و سپس بنزین را به باک اتومبیل می‌ریزد. به‌ازای هر لیتر بنزین، یک واحد به کنتور اضافه می‌شود. راننده با مشاهده این کنتور می‌فهمد که تاکنون چند لیتر بنزین وارد باک اتومبیل شده است. این کنتور یک نوع شمارنده است. پس شمارنده‌ها باید یک مقدار اولیه داشته باشند (در کنتور بنزین این مقدار صفر است). با هر بار تکرار اجرای دستورات داخل حلقه تکرار، باید مقداری به شمارنده اضافه شود (در کنتور پمپ بنزین به ازای هر لیتر بنزین یک واحد به کنتور اضافه می‌شود). تعداد دفعات اجرای حلقه تکرار با مقدار نهایی شمارنده مشخص می‌شود (در کنتور پمپ بنزین، شمارش کنتور تا پرشدن باک اتومبیل یا مقدار معینی از بنزین به دلخواه راننده ادامه می‌یابد). در حلقه تکرار نیاز به یک دستور شرطی داریم تا تشخیص دهیم که آیا عملیات موردنظر به تعداد دفعات لازم تکرار شده‌اند یا خیر. این شرط می‌تواند در ابتدا یا انتهای حلقه تکرار قرار گیرد.

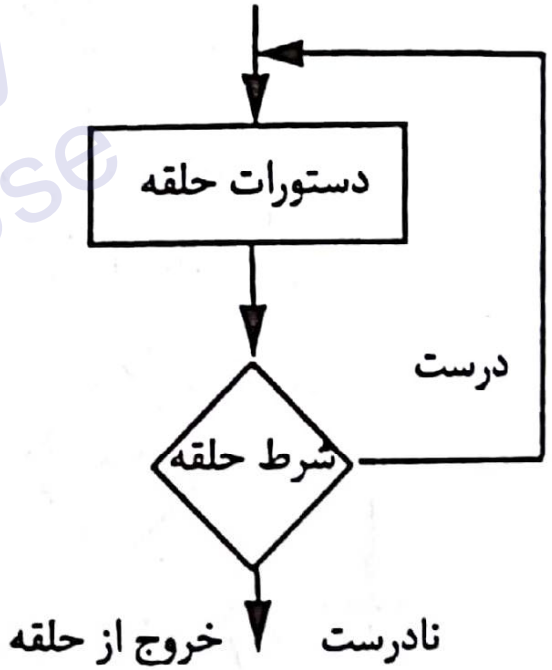
Razi University
Educational Use

حلقه تکرار در فلوچارت

در ایجاد حلقه‌های تکرار، شرط حلقه، نقش مهمی را بازی می‌کند. به طور کلی، حلقه تکرار براساس شرط حلقه بنا می‌شود. دستورات حلقه ممکن است به دو صورت در فلوچارت ظاهر شوند:



(الف)



(ب)

Razi University
Educational Use

در حالت الف، ابتدا شرط حلقه بررسی می‌شود، در صورت درست بودن شرط، دستورات حلقه اجرا می‌شوند و کنترل به ابتدای حلقه می‌رود. این روند آنقدر ادامه می‌یابد تا شرط حلقه نادرست شود. برای خاتمهٔ حلقه تکرار، شرط حلقه در داخل حلقه تکرار باید نقض شود، در غیر این صورت حلقه تکرار خاتمه نمی‌یابد.

در حالت ب، ابتدا دستورات حلقه اجرا می‌شوند و سپس شرط حلقه تست می‌گردد. اگر شرط حلقه دارای ارزش درستی باشد، کنترل الگوریتم به ابتدای حلقه می‌رود، وگرنه حلقه خاتمه می‌یابد. تفاوت این دو روش در این است که در روش ب، در صورتی که شرط حلقه درست نباشد، دستورات حلقه حداقل یک بار اجرا می‌شوند.

Razi University
Educational Use

▪ مثال ۱۷

الگوریتمی که تعداد ۵ عدد را از ورودی خوانده، اعداد زوج را در خروجی چاپ می‌کند. برای تست زوج بودن عدد، آن را بر ۲ تقسیم صحیح کرده، باقیمانده را پیدا می‌کنیم. اگر باقیمانده صفر باشد، عدد زوج است. در این الگوریتم، تعداد دفعات حلقه تکرار ۵ است که در داخل الگوریتم مشخص شده است. حلقه تکرار این الگوریتم، از دستور ۲ تا ۸ است.

Razi University
Educational Use

متغیرها

عدد مورد بررسی	X
خارج قسمت	D
باقیمانده	R
شمارنده حلقه تکرار	I

الگوریتم

۱. I ← 1

۲. تا زمانی که I <= 5 دستورات ۳ تا ۷ را تکرار کن

۳. X را بخوان

۴. D ← X / 2 (خارج قسمت صحیح)

۵. R ← X - D * 2

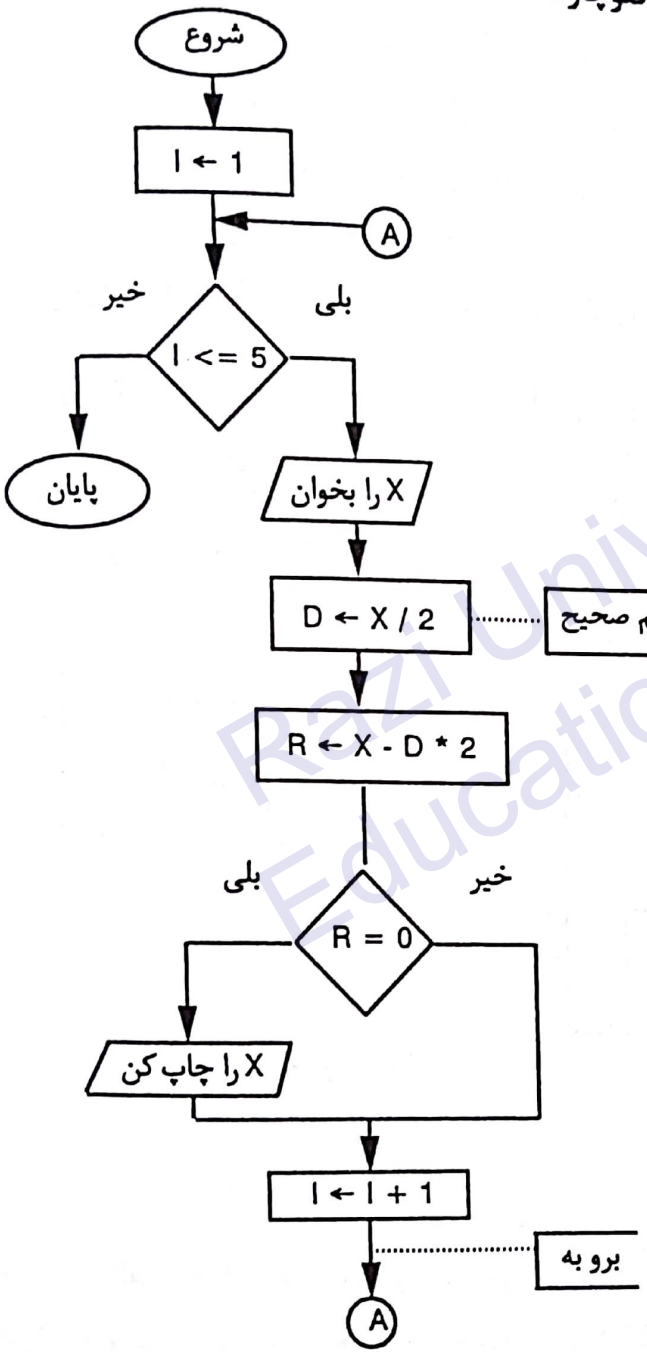
۶. اگر R = 0 آنگاه X را چاپ کن

۷. I ← I + 1

۸. پایان حلقه

۹. پایان

فلوچارت



Razi University
Educational Use

▪ مثال ۱۸

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

توضیح

در این الگوریتم، تعداد تکرار دستورات حلقه تکرار، N است که باید از ورودی خوانده شود. در مثال ۱۷ مقدار ۵ در داخل الگوریتم تعداد دفعات تکرار را مشخص می‌کرد ولی در اینجا تعداد دفعات تکرار از ورودی خوانده می‌شود. حلقه تکرار این الگوریتم از دستور ۳ تا ۷ است.

Razi University Educational Use

متغیرها

عدد خوانده شده	X
شمارنده حلقه	I
مجموع اعداد	SUM

الگوریتم

۱. N را بخوان

۲. SUM ← 0 , I ← 1

۳. تا زمانی که N => I است، دستورات ۴ تا ۶ را تکرار کن

۴. X را بخوان

۵. SUM ← X + SUM

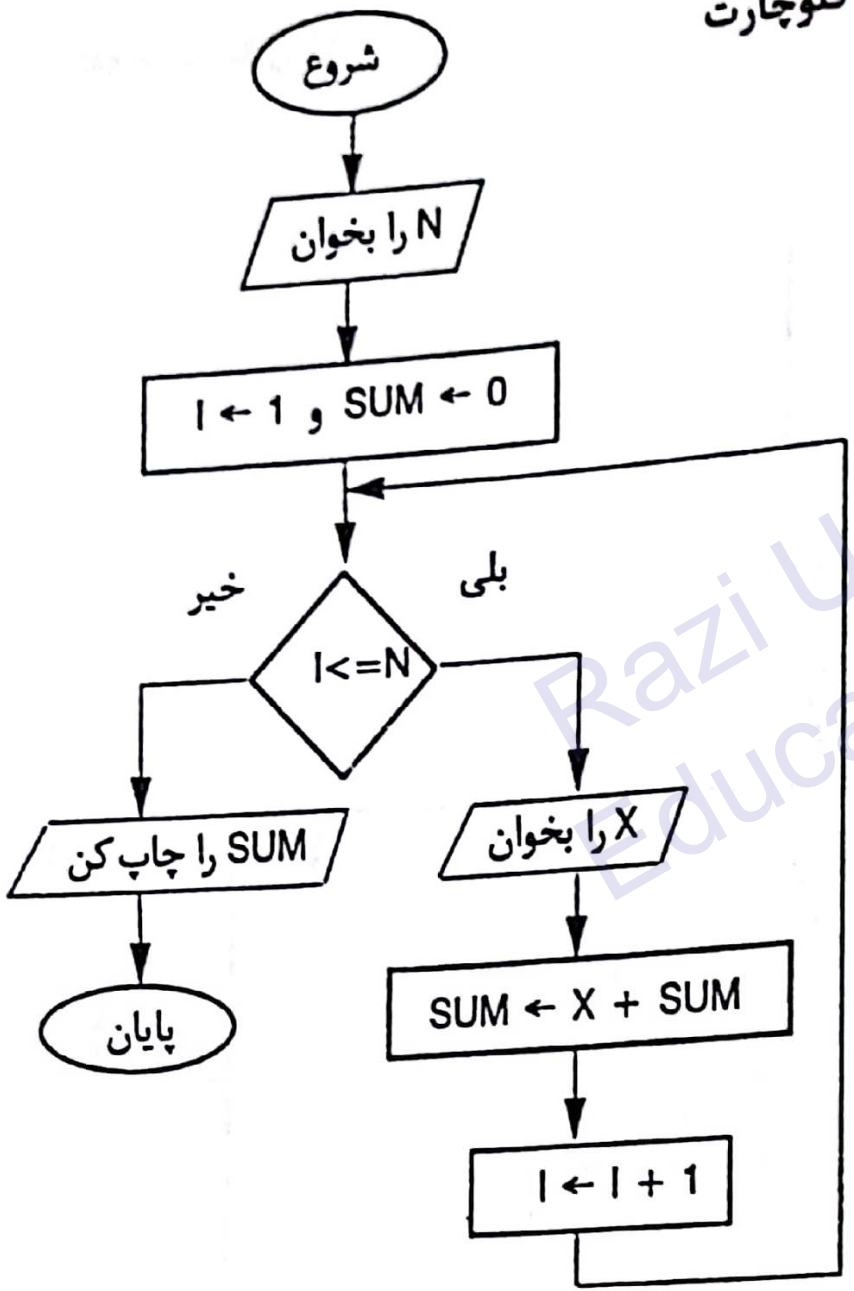
۶. I ← I + 1

۷. پایان حلقه

۸. SUM را چاپ کن

۹. پایان

فلوچارت



Razi University
Educational Use

▪ مثال ۱۹

الگوریتمی که تعداد N عدد را از ورودی خوانده، تعداد اعداد منفی، صفر و مثبت را مشخص کرده، چاپ می‌کند.

توضیح

در این الگوریتم، شرط حلقه را در پایان حلقه قرار می‌دهیم. بدین‌ترتیب، اگر شرط حلقه از همان ابتدا درست نباشد، بازهم دستورات حلقه یک بار اجرا می‌شوند.

Razi University
Educational Use

متغیرها

تعداد اعداد	N
تعداد صفر	Z
تعداد مثبت	P
تعداد منفی	NP
شمارنده حلقه	I

الگوریتم

۱. I ← 1

۲. NP ← 0 , P ← 0 , Z ← 0

۳. N را بخوان

۴. X را بخوان

۵. اگر X > 0 آنگاه P ← P + 1

وگرنه، اگر X < 0 آنگاه NP ← NP + 1

وگرنه، Z ← Z + 1

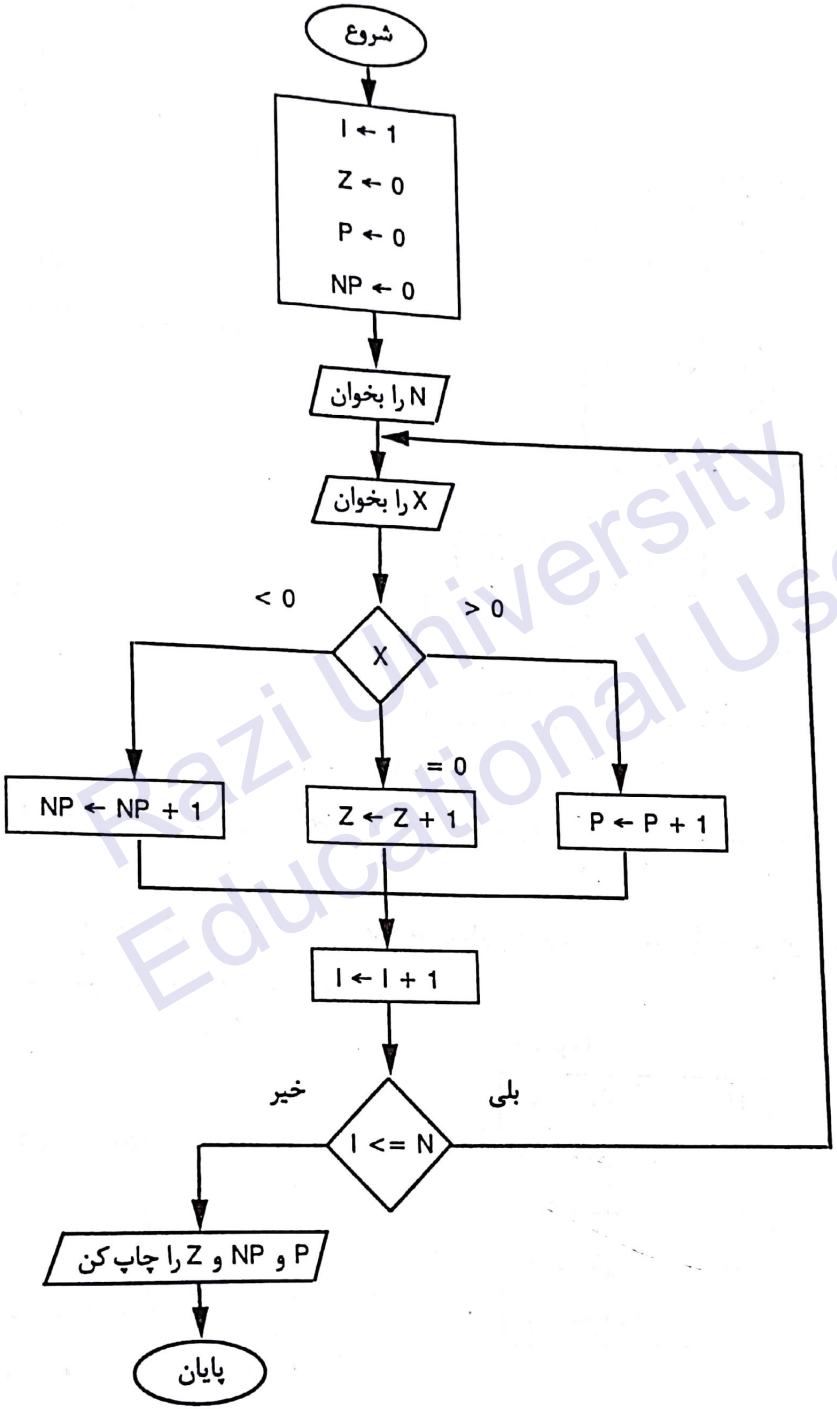
۶. I ← I + 1

۷. اگر I <= N برو به ۴

۸. P و NP و Z را چاپ کن

۹. پایان

Razi University
Educational Use



Razi University
Educational Use

▪ مثال ۲۰

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

توضیح: در این الگوریتم فرض شده است که تمام اعداد ورودی، مثبت هستند. برای پیداکردن بزرگترین عدد، ابتدا متغیر MAX برابر با اولین عدد می‌شود. سپس عددی را از ورودی خوانده، با MAX مقایسه می‌کند، چنانچه عدد ورودی، از MAX بزرگتر باشد، آن عدد در MAX قرار می‌گیرد. این روند برای تمام اعداد تکرار می‌شود و پس از حلقه تکرار، متغیر MAX در خروجی چاپ می‌شود.

Razi University
Educational Use

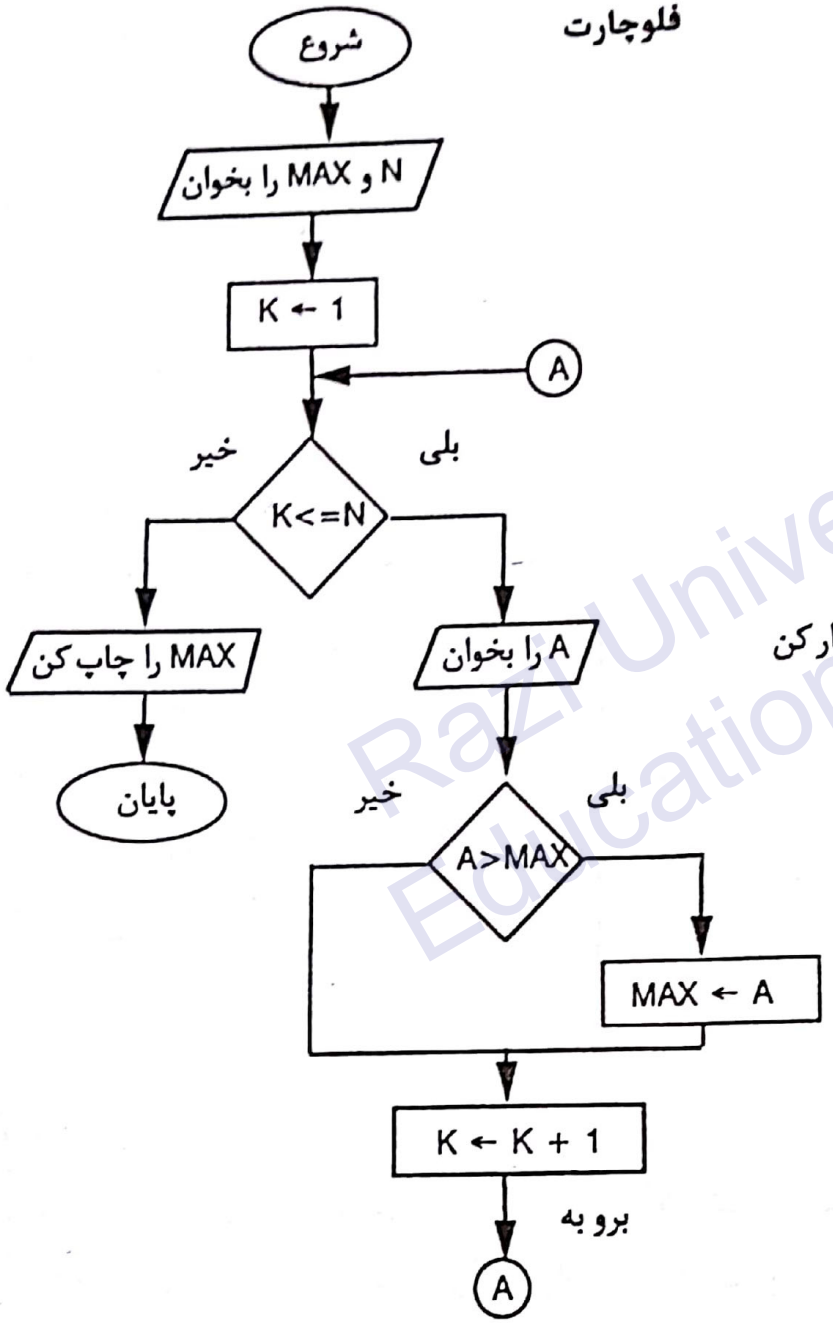
متغیرها

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

الگوریتم

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

فلوچارت



RAZI University Educational Use

▪ مثال ۲۱

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

توضیح

فاکتوریل عددی مثل N به صورت زیر محاسبه می‌شود:

$$N! = 1 * 2 * 3 * ... * (N - 1) * N$$

به عنوان مثال، فاکتوریل عدد ۴ به صورت زیر محاسبه می‌شود:

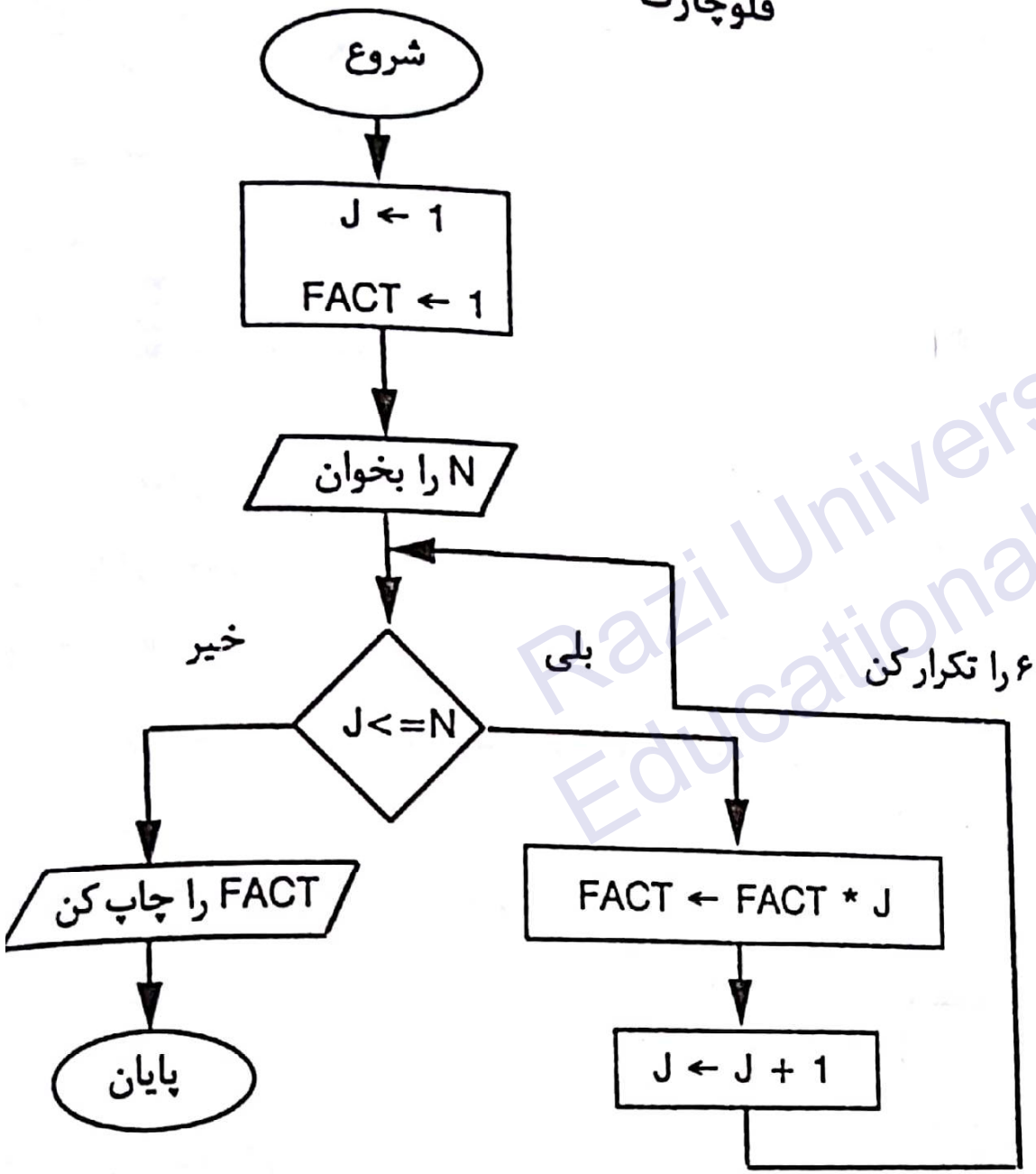
$$4! = 1 * 2 * 3 * 4$$

فاکتوریل اعداد منفی تعریف نشده است و فاکتوریل عدد صفر، برابر با یک تعریف شده است.

برای محاسبه فاکتوریل، مقدار اولیه فاکتوریل را برابر با یک تعیین می‌کنیم. سپس شمارنده‌ای را از یک تا عدد N مقدار می‌دهیم و مقدار حاصل را در فاکتوریل ضرب کرده، نتیجه جدید را نگه می‌داریم. حلقه تکرار این الگوریتم از ۴ تا ۷ است.

Razi University
Educational Use

فلوچارت



متغیرها

فاکتوریل	FACT
شمارنده	J
عدد موردنظر	N

الگوریتم

۱. J ← 1

۲. FACT ← 1

۳. N را بخوان

۴. تا زمانی که N => J است، دستورات ۵ تا ۶ را تکرار کن

۵. FACT ← FACT * J

۶. J ← J + 1

۷. پایان حلقه

۸. FACT را چاپ کن

۹. پایان

Razi University
Educational Use

▪ مثال ۲۲

الگوریتمی که نام، ساعات کار و دستمزد ساعتی کارکنان مؤسسه‌ای را خوانده، حقوق آنها را محاسبه می‌کند. اگر کارمندی بیش از ۴۰ ساعت کار کرده باشد، اضافه کار به او تعلق می‌گیرد. به ازای هر ساعت اضافه کاری، $\frac{۳}{۲}$ دستمزد ساعتی، به عنوان اضافه کاری پرداخت می‌شود. حلقه تکرار این الگوریتم از دستور ۳ تا ۸ است.

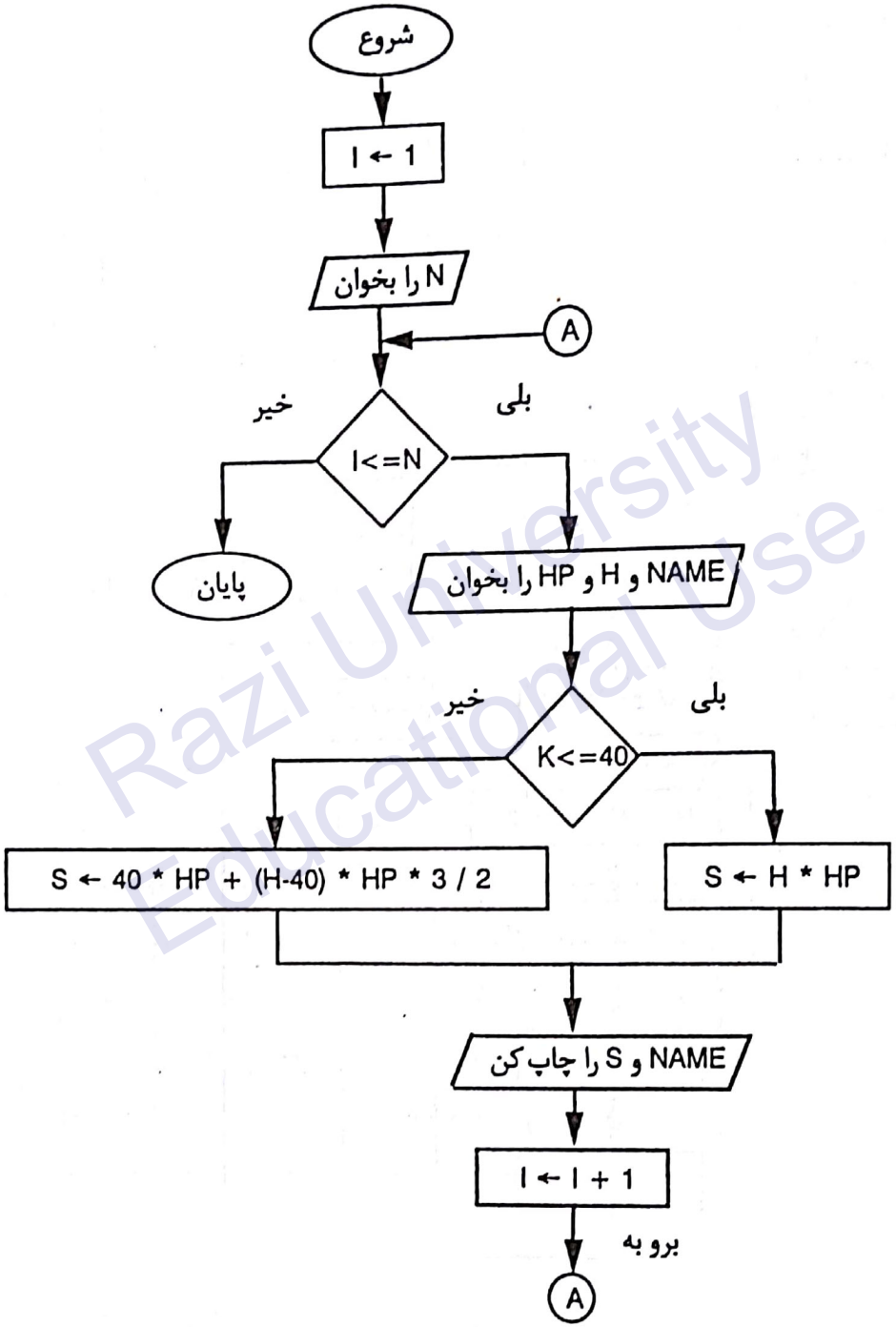
متغیرها

تعداد کارکنان	N
نام	NAME
ساعت کارکرد	H
ساعتی	HP
اضافه‌کاری	OV
شمارنده	I
حقوق	S

الگوریتم

۱. I ← 1
۲. N را بخوان
۳. تا زمانی که N => I است، دستورات ۴ تا ۷ را اجرا کن
۴. NAME ، H و HP را بخوان
۵. اگر H <= 40 آنگاه S ← H * HP
وگرنه S ← 40 * HP + (H-40) * HP * 3/2
۶. NAME و S را چاپ کن
۷. I ← I + 1
۸. پایان حلقه
۹. پایان

Razi University Educational Use



Razi University
Educational Use

▪ مثال ۲۳

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

توضیح

تبدیل یک عدد از مبنای ۱۰ به ۲، با تقسیمهای متوالی انجام می‌شود و باقیمانده‌ها از آخرین باقیمانده به اولین باقیمانده، نوشته می‌شوند. برای این منظور شمارنده‌ای در نظر گرفته شد که هر بار، ۱۰ به توان آن شمارنده می‌رسد و در باقیمانده ضرب می‌شود و حاصل کار با هم جمع می‌شوند. پس از مشاهده الگوریتم، آن را برای تبدیل عدد ۳ از مبنای ۱۰ به مبنای ۲، امتحان می‌کنیم.

Razi University Educational Use

متغیرها

عدد مبنای ۱۰	N
شمارنده	I
خارج قسمت تقسیم بر ۲	M
باقیمانده	R
عدد مبنای ۲	S

الگوریتم

۱. $S \leftarrow 0$

۲. $I \leftarrow 0$

۳. N را بخوان

۴. تا زمانی که $N \neq 0$ 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. $I \leftarrow I + 1$

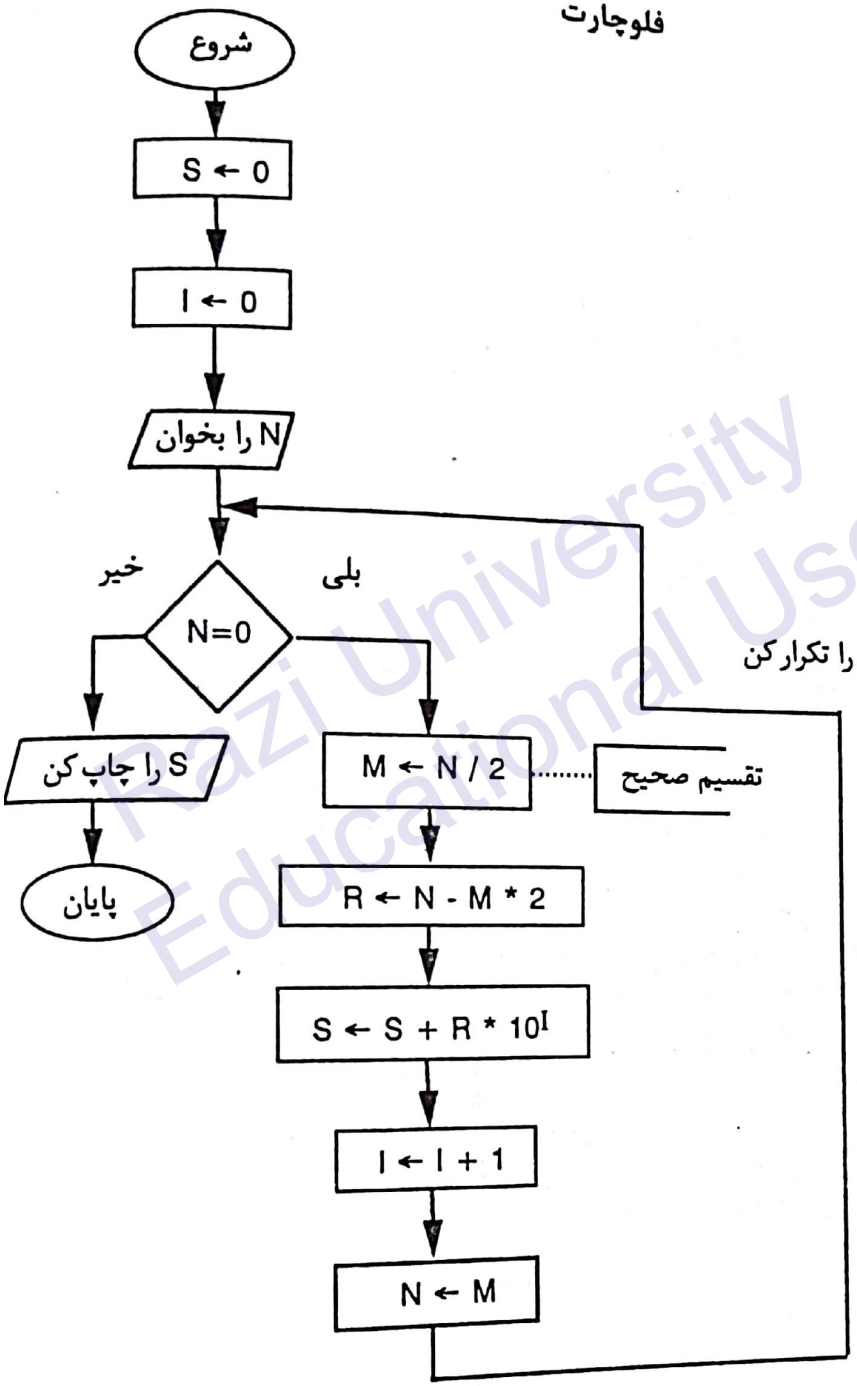
۹. $N \leftarrow M$

۱۰. پایان حلقه

۱۱. S را چاپ کن

۱۲. پایان

فلوچارت



Razi University
Educational Use

شماره دستور	N	I	M	R	S
1					
2		0			0
3	3				
4					
5			1		
6				0	
7					1
8		1			
9	1				
10					
4					
5			0		
6				1	
7					11
8		2			
9	0				
10					
4					
11					
12					

وقتی الگوریتم را به ازای N=3 امتحان کردیم، مقدار ۱۱ در S قرار گرفت که نتیجهٔ مورد انتظار ماست. بنابراین، الگوریتم به درستی عمل می‌کند.

Razi University
Educational Use

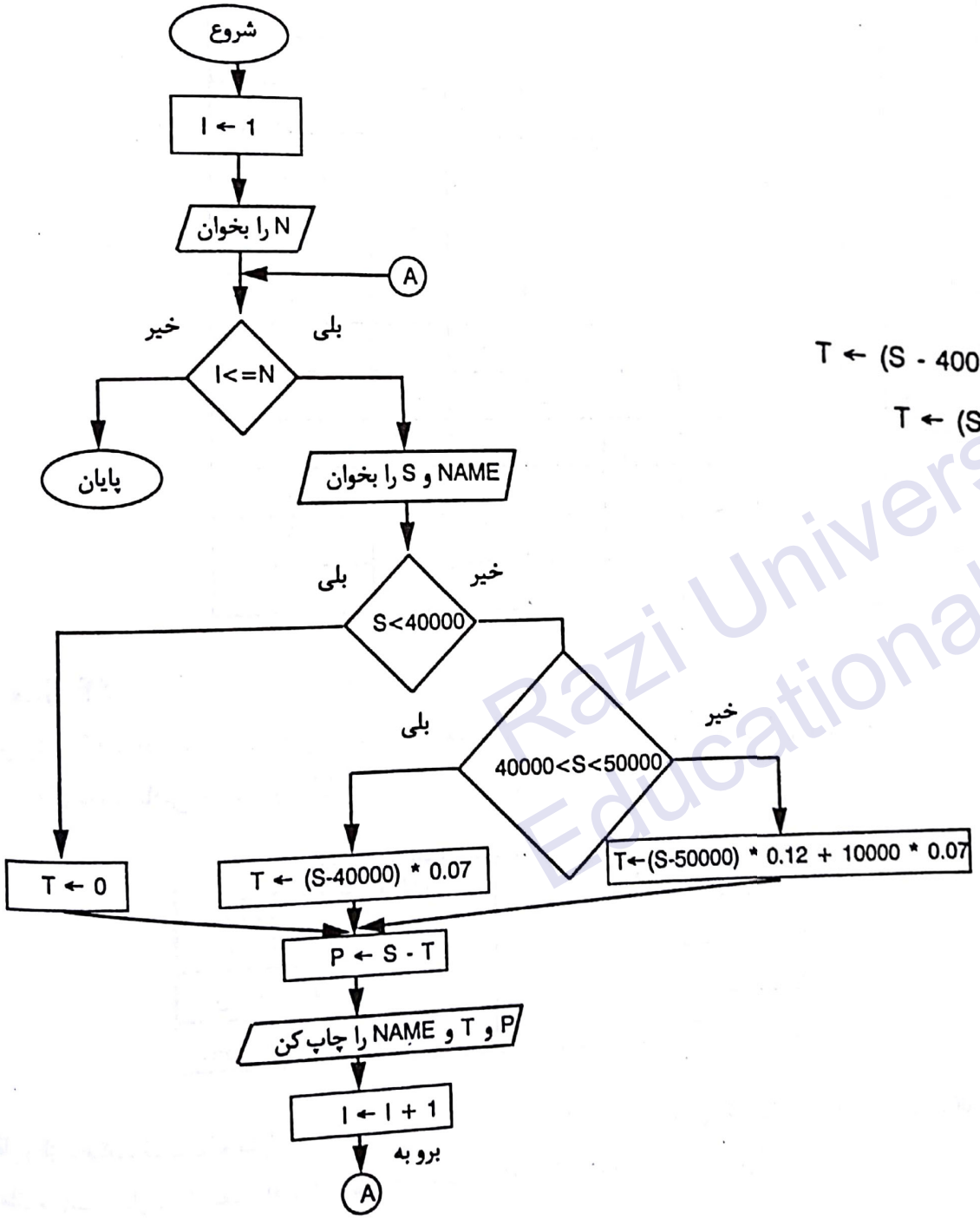
▪ مثال ۲۴

الگوریتمی که درآمد ماهانه تعدادی کارمند را به همراه نام آنها می‌خواند و براساس تعرفه زیر مالیات آنها را محاسبه کرده، دریافتی خالص و مالیات را به خروجی می‌برد.

درآمد ماهانه (تومان)	مالیات
کمتر از 40,000	0
بین 40,000 تا 50,000	۷٪ نسبت به مازاد
بیش از 50,000	۱۲٪ نسبت به مازاد

منظور از مالیات نسبت به مازاد این است که، به عنوان مثال، اگر حقوق کسی 45,000 تومان باشد، 40,000 تومان معافیت دارد و از بقیه (45,000 - 4,000) که مازاد نام دارد، ۷٪ مالیات کسر می‌گردد.

Razi University Educational Use



الگوریتم

۱. I ← 1

۲. N را بخوان

۳. تا زمانی که N => I دستورات ۴ تا ۸ را اجرا کن

۴. NAME و S را بخوان

۵. اگر S < 40000 آنگاه T ← 0

وگرنه اگر S > 40000 و S < 50000 آنگاه T ← (S - 40000) * 0.07

وگرنه T ← (S - 50000) * 0.12 + (50000 - 40000) * 0.07

۶. P ← S - T

۷. P ، T و NAME را چاپ کن

۸. I ← I + 1

۹. پایان حلقه

۱۰. پایان

متغیرها

تعداد کارکنان	N
شمارنده	I
مالیات	T
حقوق	S
دریافتی خالص	P
نام	NAME

Razi University
Educational Use

- مثال ۲۵

الگوریتمی که ۲۰ جمله اول دنباله فیبوناچی را تولید و چاپ کند.

Razi University
Educational Use

توضیح

در این دنباله، دو جمله اول، یک است و جملات بعدی از مجموع دو جمله قبلی به دست می‌آید:

1, 1, 2, 3, 5, 8, 13, ...

جملات اول و دوم را F1 و F2 و جمله‌ای که باید تولید شود را F3 می‌نامیم. بدین ترتیب، F3 = F1 + F2.
سپس محتویات آنها را برای تولید جمله بعدی عوض می‌کنیم.

متغیرها

شمارنده	K
جملات دنباله	F1, F2, F3

الگوریتم

۱. F1 ← 1

۲. F2 ← 1

۳. F1 و F2 را چاپ کن

۴. K ← 3

۵. تا زمانی که K <= 20 است، دستورات ۶ تا ۱۰ را اجرا کن

۶. F3 ← F1 + F2

۷. F3 را چاپ کن

۸. F1 ← F2

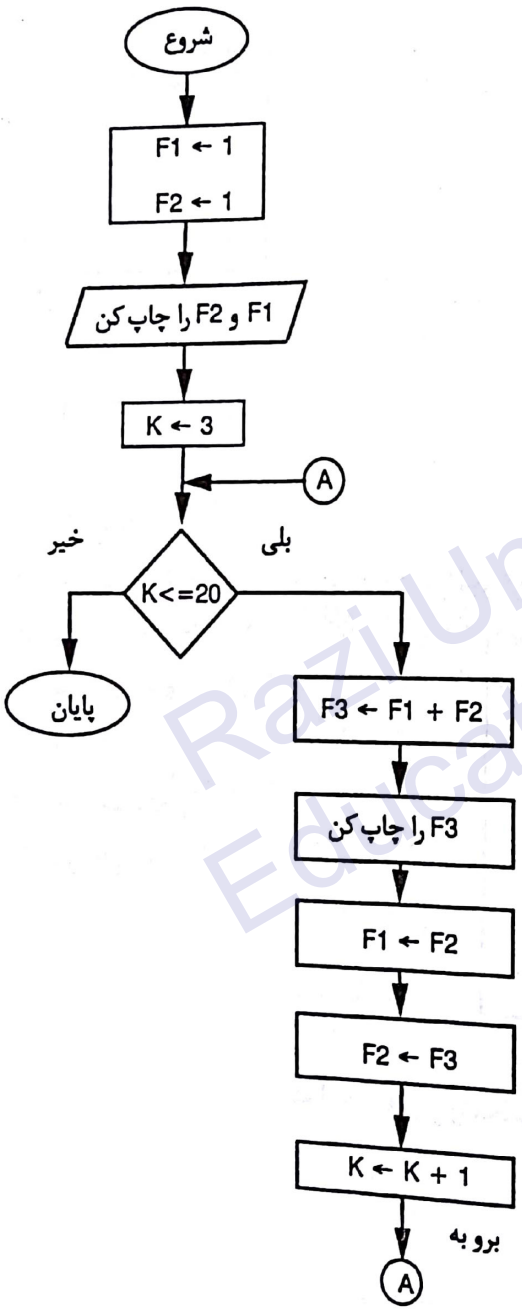
۹. F2 ← F3

۱۰. K ← K + 1

۱۱. پایان حلقه

۱۲. پایان

فلوچارت



Razi University Educational Use