فصل اول

روش حل مسئله

1- خواندن دقیق مسئله با جزئیات کامل
2- بیرون کشیدن فرضیات مسئله از داخل مسئله
3- ارتباط بین این فرضیات و انتخاب روش حل
4- پیاده سازی به وسیله یک زبان برنامه نویسی

مثال :

خرید بلیط از یک آژانس هواپیمایی :

1- شروع .
2- مسافر وارد آژانس مسافرتی می شود .
3- کارمند بخش مربوط به فروش ، چک می کند برای مقصد مورد نظر جا وجود دارد یا نه
4- اگر جا موجود نباشد به مسافر جواب داده می شود که جا موجود نیست .
5- اگر جا موجود باشد فرم مربوط به مشخصات مسافر پر می شود .
6- بلیط تهیه شده و در اختیار مسافر قرار داده می شود
7- پایان.

همانطوری که مشاهده می شود ، مسافر باید جهت خرید بلیط یک سری مراحل را به ترتیب طی کند. تا نهایتا بلیط برای او صادر شود. این مراحل که از یک نقطه آغاز می شود و در یک نطقه به پایان می رسد . این مراحل در زبان برنامه نویسی **الگوریتم** می گویند.

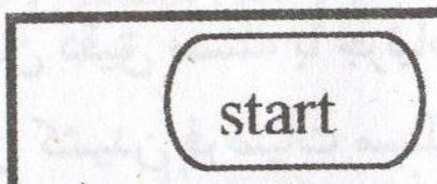
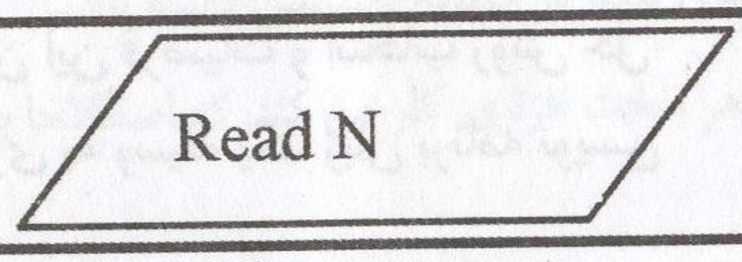
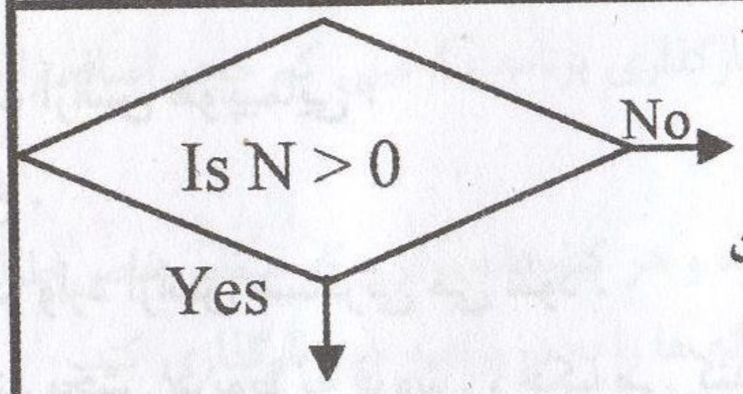
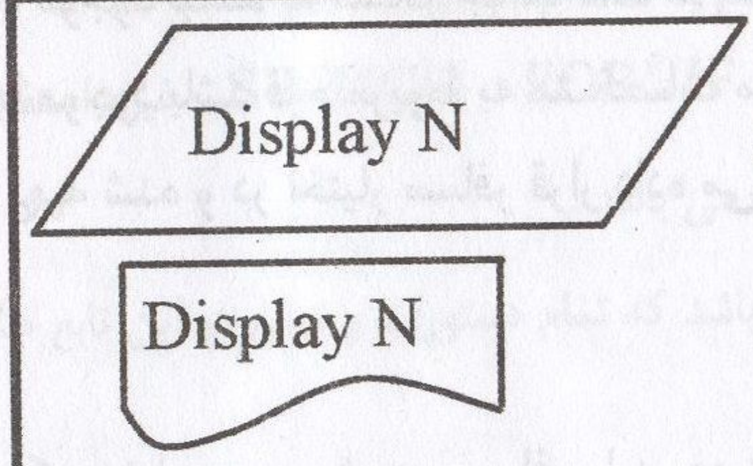
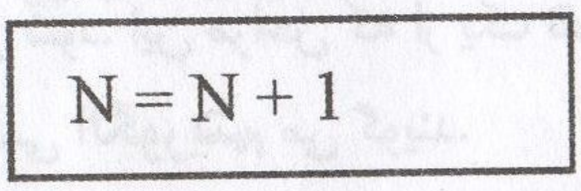
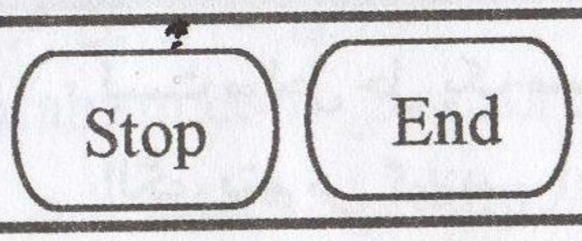
لیست مراحل حل یک مسئله را که بوسیله یک زبان بیانی ،نوشته می شود را **الگوریتم** می گویند.

مراحل حل نموداری یک مسئله به وسیله نمادهای بخصوصی که هر کدام بیانگر یک عمل می باشند **فلوچارت** می گویند

مزایایی استفاده از فلوچارت :

- فهم مسئله را آسان می کند که ازچه مراحلی (Steps) تشکیل شده است .
- قادر می سازد که برنامه نویس برنامه را اشکال زدایی (Debug) کند.
- قارد می سازد که برنامه نویس برنامه را ردیابی (Trace) کند.
- هر کدام از نمادها به همدیگر متصل شده مراحل را قدم به قدم بصورت بصری نمایش می دهند.

نمادهایی که در رسم فلوچارت استفاده می شوند :

1	**شروع:** برنامه با این نماد شروع می شود . 
2	**ورودی** :با استفاده ازاین نماد کاربر می تواند مقادیر را وارد کند. 
3	**شرط :** این شرط دارای دو شاخه Yes و No می باشد اگر شرط درست باشد به شاخه Yes می رود اگر شرط نادرست باشد به شاخه No می رود . 
4	**خروجی** :پیغام ها یا و متغیرها می توانند در کدام از این دو نماد قرارد گیرند. 
5	**انتساب و پردازش:** جهت انتساب و و پردازش از این نماد استفاده میشود. 
6	**پایان :** این نماد پایان برنامه نشان می دهد. 
7	**ارتباط** :برای ارتباط دادن خطوط بکار می رود . این نماد نشان می دهد ادامه این مرحله در کجا قرار دارد . 
8	**خطوط ارتباطی:** جهت حرکت را نشان می دهد . 

دانشجویان عزیز توجه نمایید برای کدام از مراحل الگوریتم یک شکل دایره ای شکل در جلوی هر نماد فلوچارت کشیده شده است که هر کدام از این شکلهای دایره ای شکل نشانده هر مرحله از الگوریتم می باشد.

مسئله ۱ :

برنامه‌ای بنویسید که یک عدد از ورودی بخواند آن را در خروجی نمایش دهد .

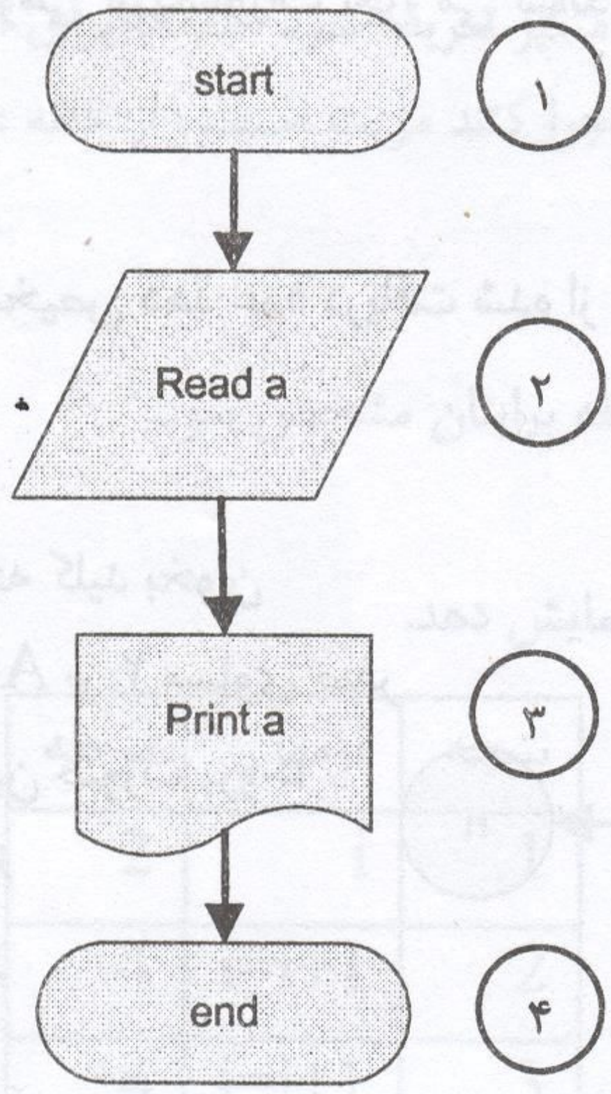
الگوریتم ، فلوچارت

۱- شروع .

۲- بخوان متغیر A از ورودی.

۳- متغیر A در خروجی چاپ کن .

۴- پایان .



متغیر : به مکانی از حافظه گفته می شود که می توان در آن یک مقدار ذخیره یا بازیابی کرد . اگر مقدار جدیدٍ در آن ریخته شود مقدار قبلی از بین خواهد رفت .

با استفاده از نمادهای خوانده شده کاربر می تواند مسئله را بصورت بصری رسم کند باید به این نکته توجه داشته باشیم که تمام برنامه ها از یک نقطه شروع و یک نقطه پایان دارند .

مسئله ۲ :

برنامه ای بنویسد که دو عدد از صفحه کلید خوانده کامپیوتر مجموع آنها را حساب کند.

الگوریتم و فلوچارت

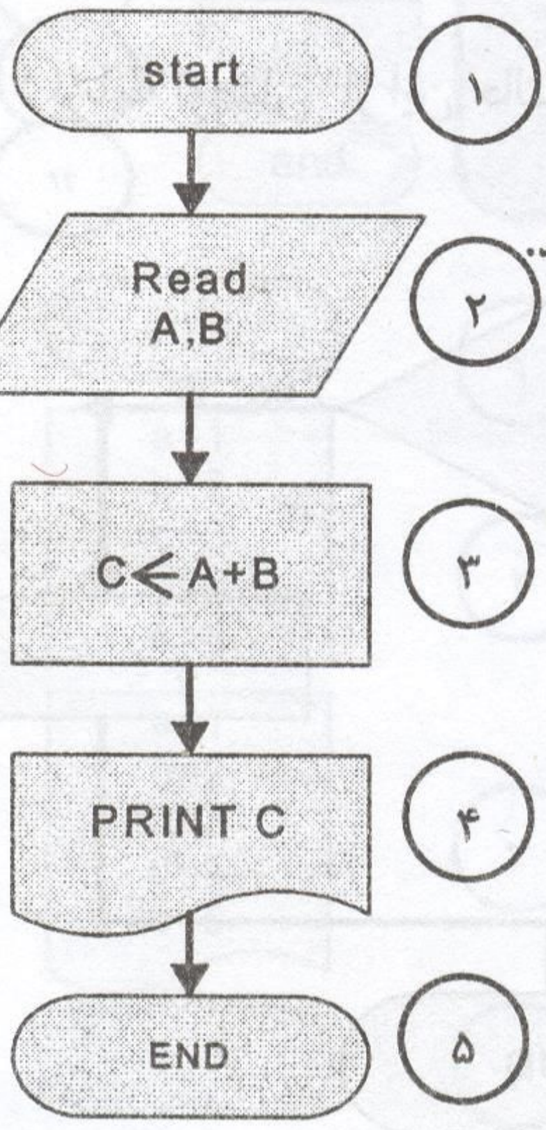
۱- شروع

۲- دو متغیر A,B را بخوان

۳- دو متغیر A,B را باهم جمع کن و در متغیر C قرارد بده

۴- متغیر C را چاپ کن

۵- پایان .



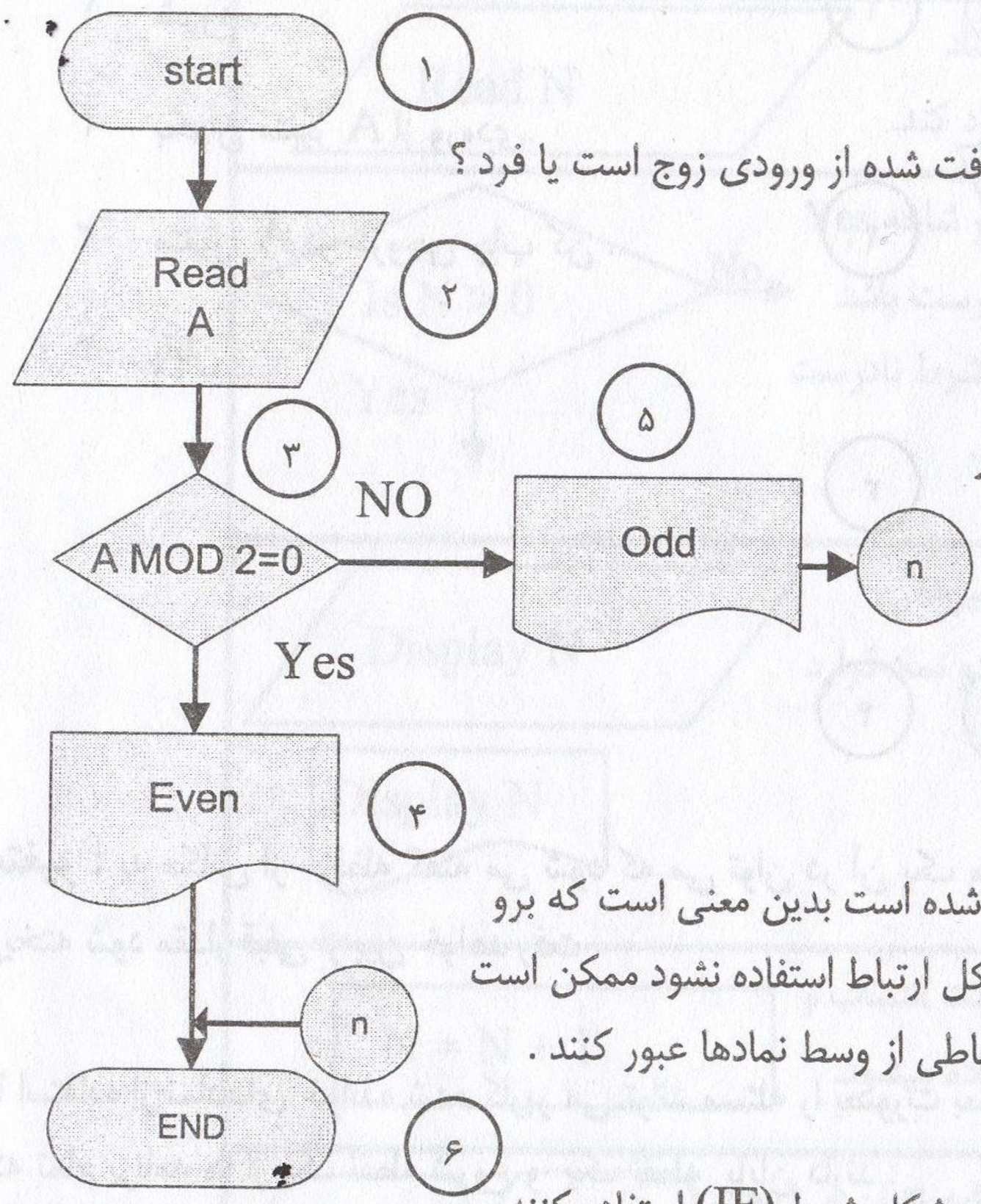
در این مسئله علمیات محاسبات در شکل مستطیل انجام شده است .

ساختار کنترلی if.....else

از ساختار کنترلی if...else زمانی استفاده می شود دو حالت درست بودن شرط و نبودن آن بررسی شود اگر شرط درست باشد یک دستور یا گروهی از دستورات اجرا می شوند و در غیر این صورت یک دستور یا گروهی از دستورات اجرا می شوند.

مسئله ۳ :

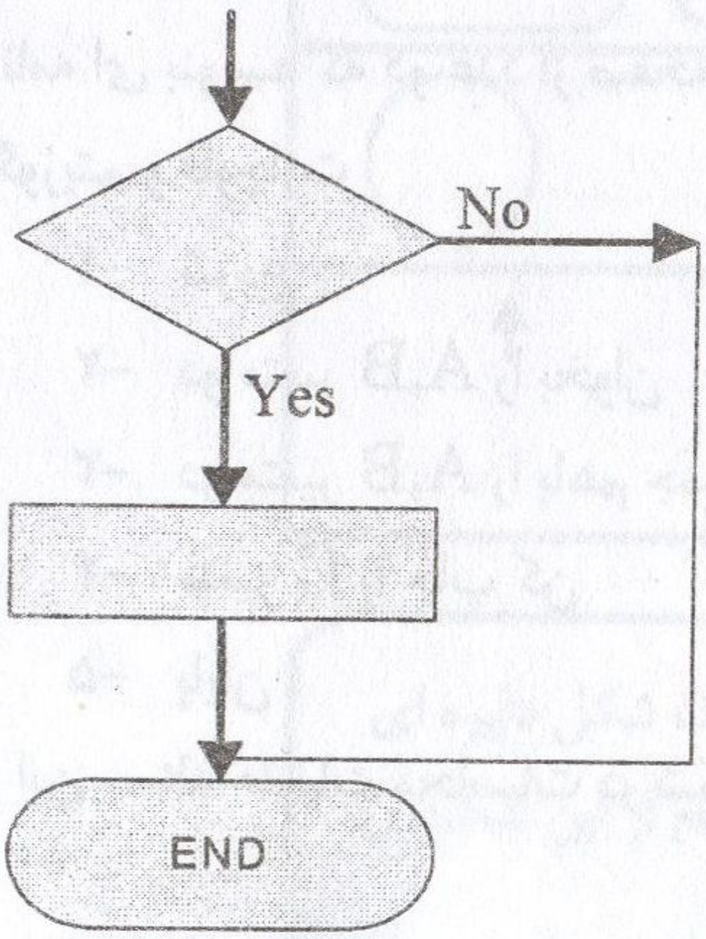
برنامه ای بنویسید کامپیوتر تشخیص دهد عدد دریافت شده از ورودی زوج است یا فرد ؟

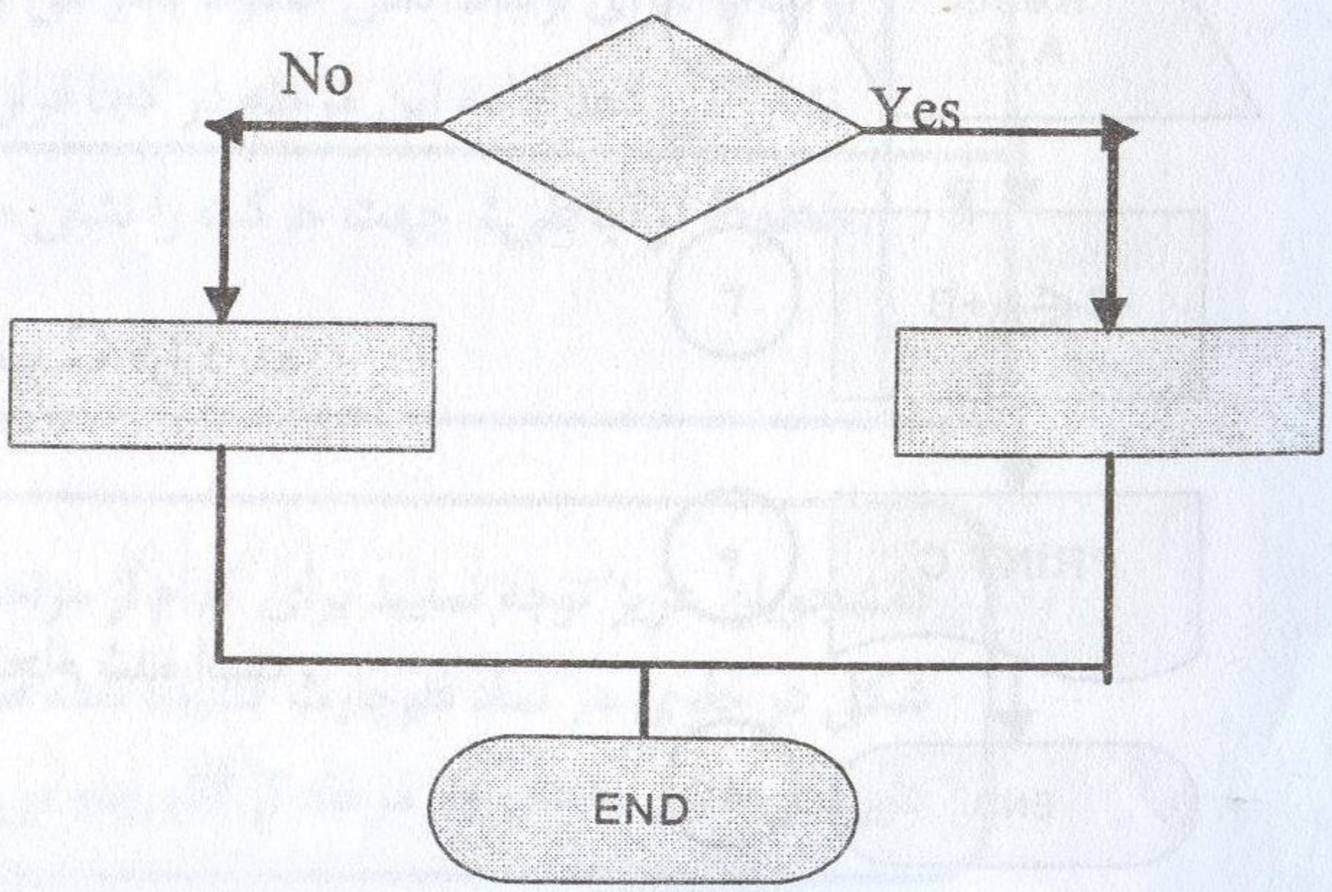


۱- شروع
۲- متغیر A را از صفحه کلید بخوان
۳- اگر باقیمانده متغیر A بر ۲ مساوی صفر باشد برو ۴ در غیراین صورت برو ۵
۴- چاپ کن زوج برو ۶
۵- چاپ کن فرد برو ۶
۶- پایان .

در این مثال از شکل ارتباط با برچسب n استفاده شده است بدین معنی است که برو جای که این برچسب n درج شده است اگر از شکل ارتباط استفاده نشود ممکن است فلوچارت سادگی خود را از دست دهد و خطوط ارتباطی از وسط نمادها عبور کنند .

نکته : برنامه نویسان از این حالتها می توانند برای شکل شرط (IF) استفاده کنند.





آشنایی با ساختارهای کنترلی حلقه

در بعضی از برنامه‌ها احتیاج است که یکسری از دستورات به بصورت مکرر تکرار شوند تا یک پرسه خاص انجام پذیر
یا برای نوشتن بعضی از برنامه‌ها احتیاج است تعداد زیادی متغیر تعریف شود حلقه‌ها برای این بوجود آمدند که یکسری
از دستورات را بصورت مکرر تا رسیدن به یک شرط خاص اجرا کنند مزیت استفاده از حلقه :

۱- جلوگیری از نوشتن دستورات تکراری

۲- جلوگیری از تعریف متغیرهای اضافی

به مسئله‌های زیر توجه نمایید تا مفهموم ساختار کنترلی حلقه برایتان مشخص شود.

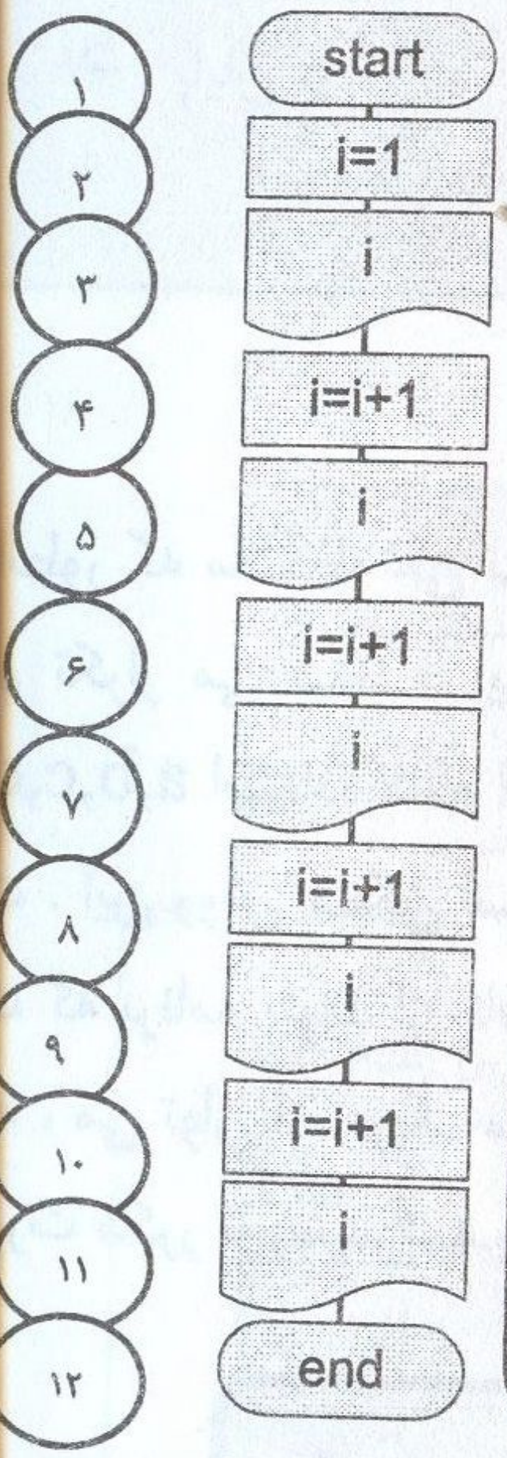
مسئله ٤ :

برنامه‌ای بنویسید که اعداد ۱ تا ۵ را کامپیوتر در خروجی نمایش دهد.

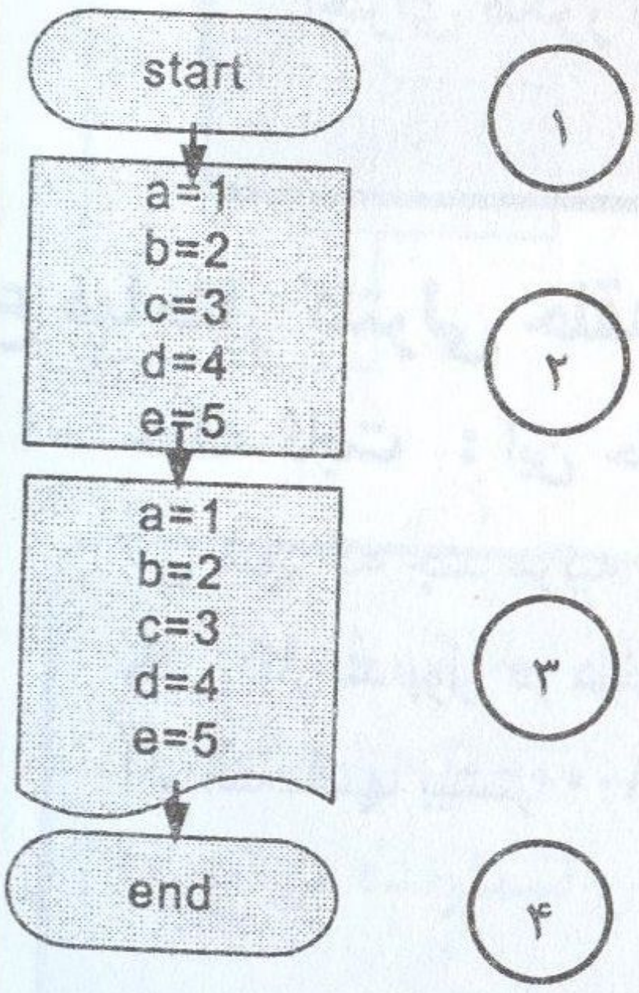
روش اول : بدون حلقه

۱- شروع
۲- به متغیر i مقدار یک را نسبت بده
۳- متغیر i را چاپ کن
۴- یک واحد به متغیر i اضافه کن (i=i+1)
۵- متغیر i را چاپ کن
۶- یک واحد به متغیر i اضافه کن (i=i+1)
۷- متغیر i را چاپ کن
۸- یک واحد به متغیر i اضافه کن (i=i+1)
۹- متغیر i را چاپ کن
۱۰- یک واحد به متغیر i اضافه کن (i=i+1)
۱۱- متغیر i را چاپ کن
۱۲- پایان.

نتیجه	فرمول	مرحله
1	i	2
2	i=i+1	4
3	i=i+1	6
4	i=i+1	8
5	i=i+1	10



هر بارکه متغیر i مقدار جدیدی را دریافت کند مقدار قبلی آن از بین خواهد رفت .

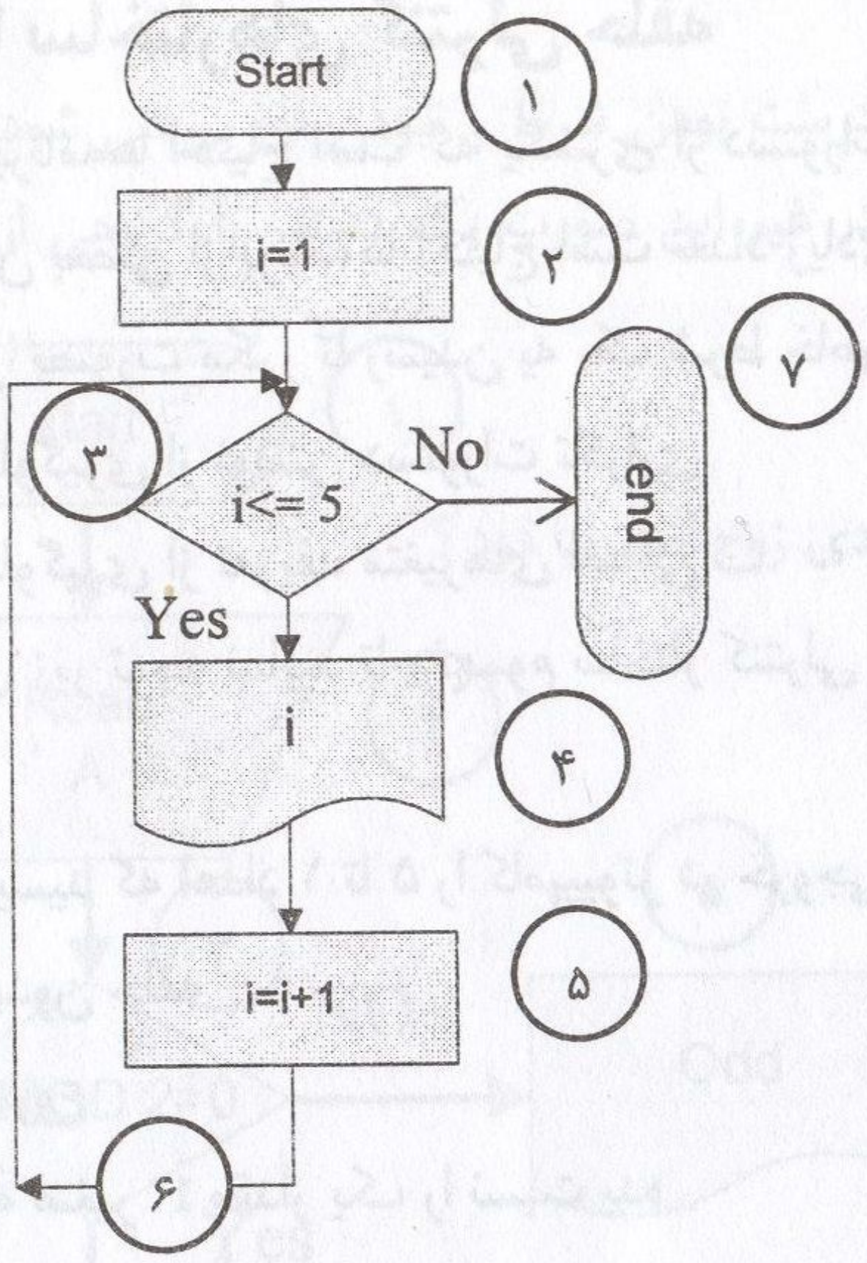


روش دوم : استفاده از متغیر جداگانه

۱- شروع
۲- مقادیر ۱ تا ۵ را بترتیب در متغیرهای a,b,c,d,e انتساب دهید .
۳- متغیر را چاپ کن
۴- پایان .

روش سوم : برنامه با استفاده از حلقه

۱- شروع
۲- به متغیر i مقدار یک را نسبت بده
۳- اگر مقدار متغیر i کوچکتر ۵ باشد برو به ۴
در غیراین صورت به برو ۷
۴- متغیر i را چاپ کن
۵- یک واحد به متغیر i اضافه کن (i=i+1)
۶- برو به مرحله ۳
۷- پایان .



همانطور که ملاحظه می نمایید در روش اول مسئله چهارم برنامه در ۱۲ مرحله نوشته شد و یک سری از کدها پشت سرم تکرار می شدند . این تکرار باعث افزایش کدهای[1] تکراری در برنامه می شود. در روش دوم از ۵ متغیر a,b,c,d,e استفاده شده است و به طور مجزا مقادیر ۱ تا ۵ به هرکدام از متغیرهای a,b,c,d,e نسبت داده شده است . این روش ممکن است ساده باشد ولی حجم حافظه را خیلی اشغال می کند و یکی از مزایای برنامه خوب این است که برنامه بتواند از حافظه به بهینه‌ترین شکل استفاده کند . در روش سوم با استفاده از یک متغیر و بوسیله یک حلقه ، می توان از تعریف متغیرهای اضافی و نوشتن کد های تکراری جلوگیری کرد. با این روش یکسری از دستورات بصورت مکرر تا رسیدن به یک شرط خاص تکرار می شوند.

اجرای مکرر دستورات تا رسیدن به یک شرط خاص را حلقه می گویند

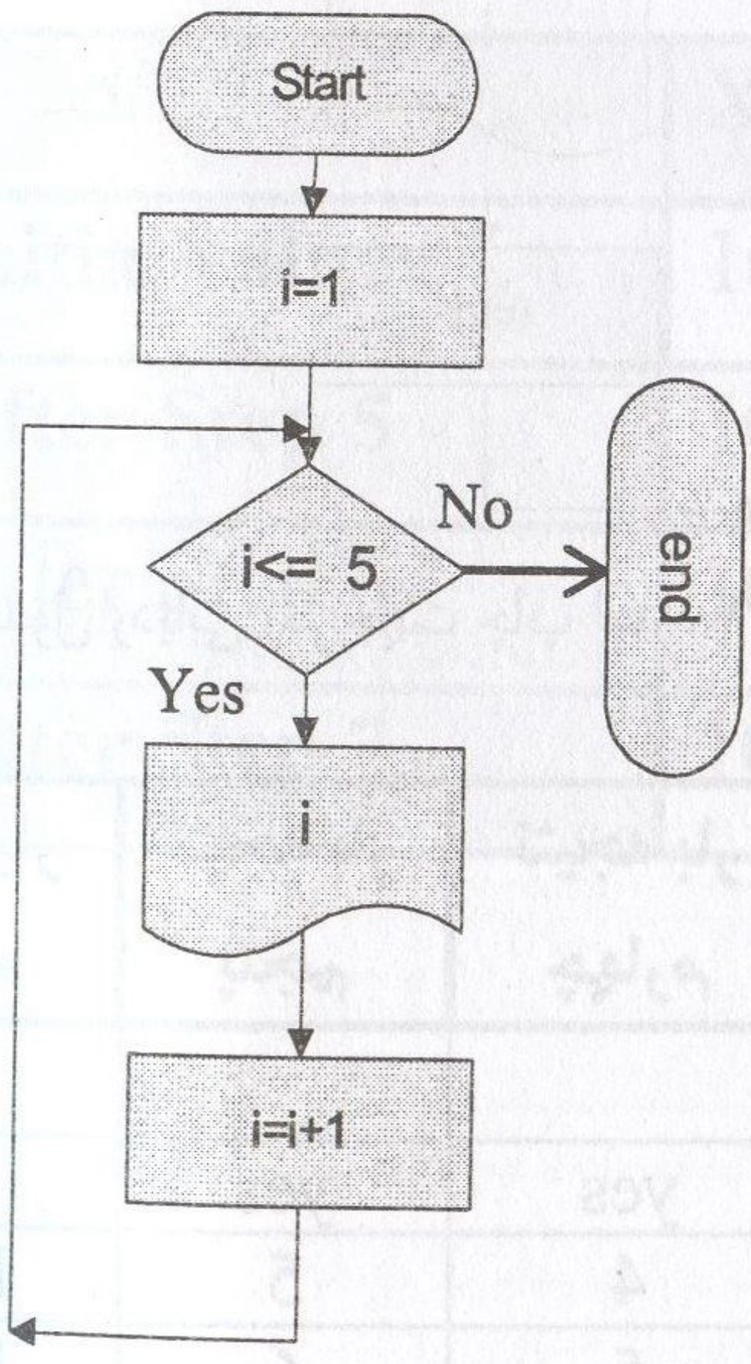
انواع ساختار کنترلی حلقه

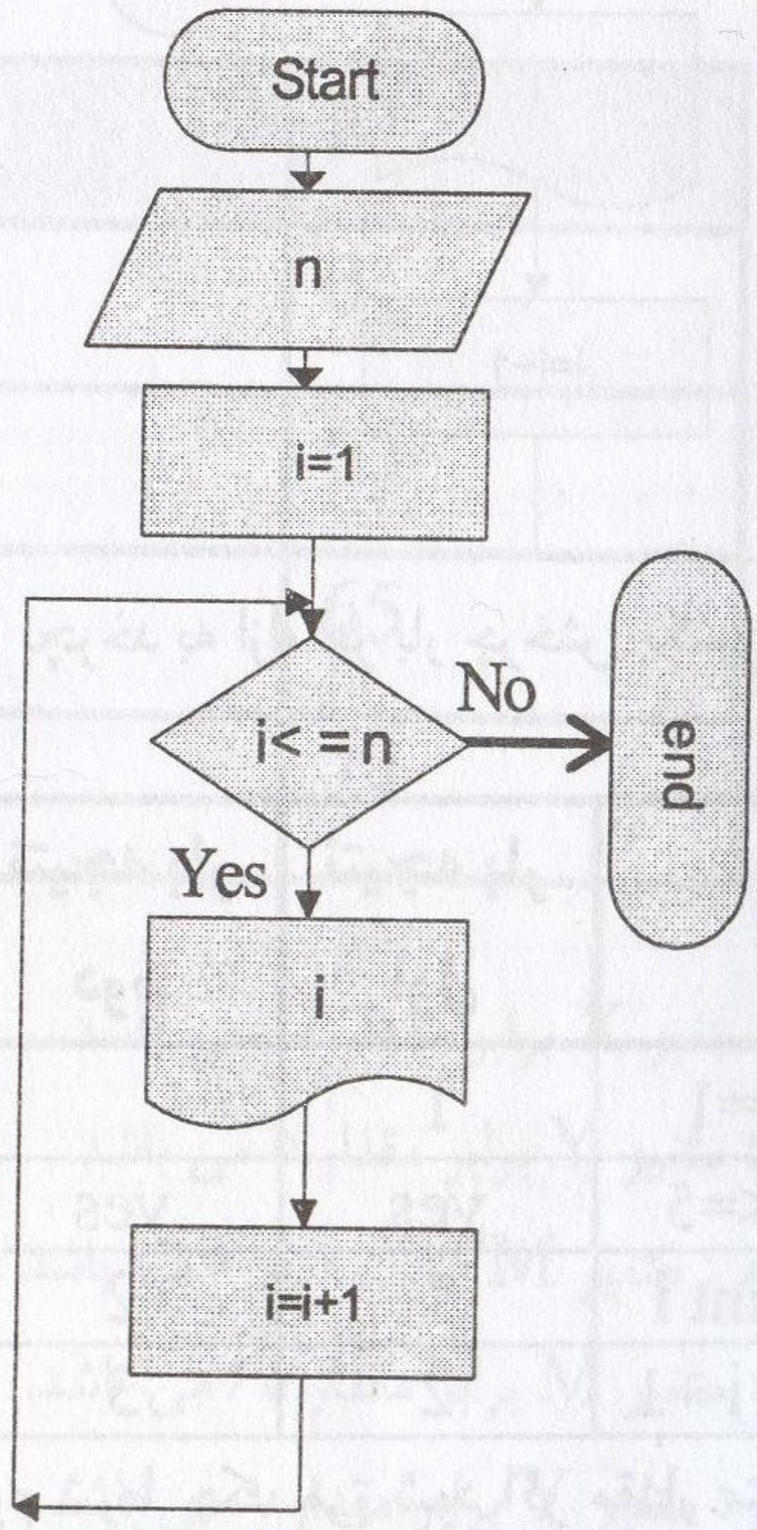
۱- **حلقه ثابت :** این حلقه به تعداد مشخصی دستورات را تکرار می‌کند و توسط برنامه نویس مشخص می شود که چند مرتبه باید حلقه دستورات را تکرار کند. مثل دستگاه پول شمار در بانک یا در فر وشگاه ، وقتی کارمند پول در داخل دستگاه قرار می دهد دستگاه فقط می تواند اسکناسها را تا عدد ۱۰۰ شمارش کند اگر اسکناسها بیشتر ۱۰۰ فقره باشند دستگاه آنها را شمارش نمی‌کند.

۲- **حلقه متغیر :** در این نوع حلقه تعداد دفعات تکرار دستورات بستگی به عدد وارد شده از صفحه کلید دارد. در واقع این حلقه توسط برنامه نویس طوری طراحی شده است که حلقه به تعداد عدد وارد شده از ورودی ، تکرار شود بدین صورت که کاربر هر عددی را وارد کند حلقه به تعداد عدد وارد شده تکرار می‌شود . مثال خوبی که برای این نوع حلقه می‌توان زد دستگاه عابر بانک می‌باشد. وقتی کاربر شماره کاربری و کلمه عبور را می دهد دستگاه منتظر می‌ماند تا کاربر مبلغ درخواستی خود را وارد کند تا به اندازه آن مبلغ اسکناسهای ۱۰۰۰ تومانی یا ۲۰۰۰ تومانی پرداخت نماید.

حلقه متغیر : حلقه به تعداد عدد n که از صفحه کلید وارد می شود تکرار می شود.

حلقه ثابت : فقط ۵ مرتبه تکرار می شود.

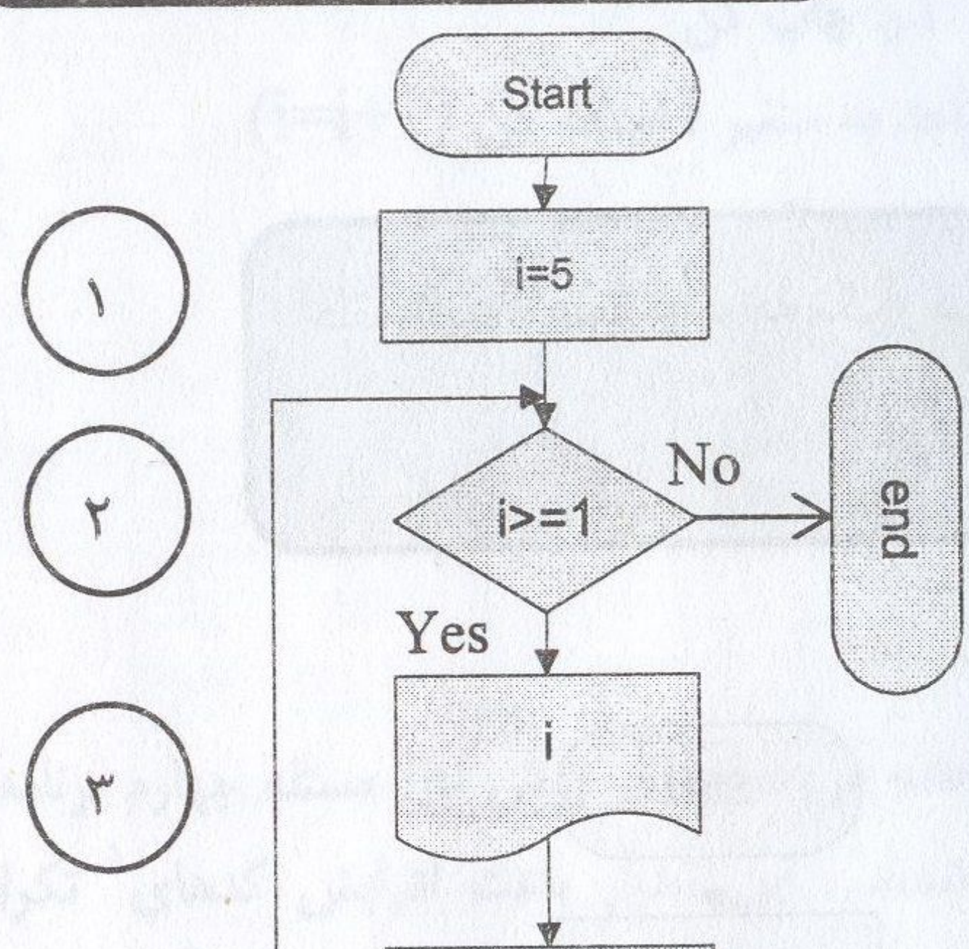




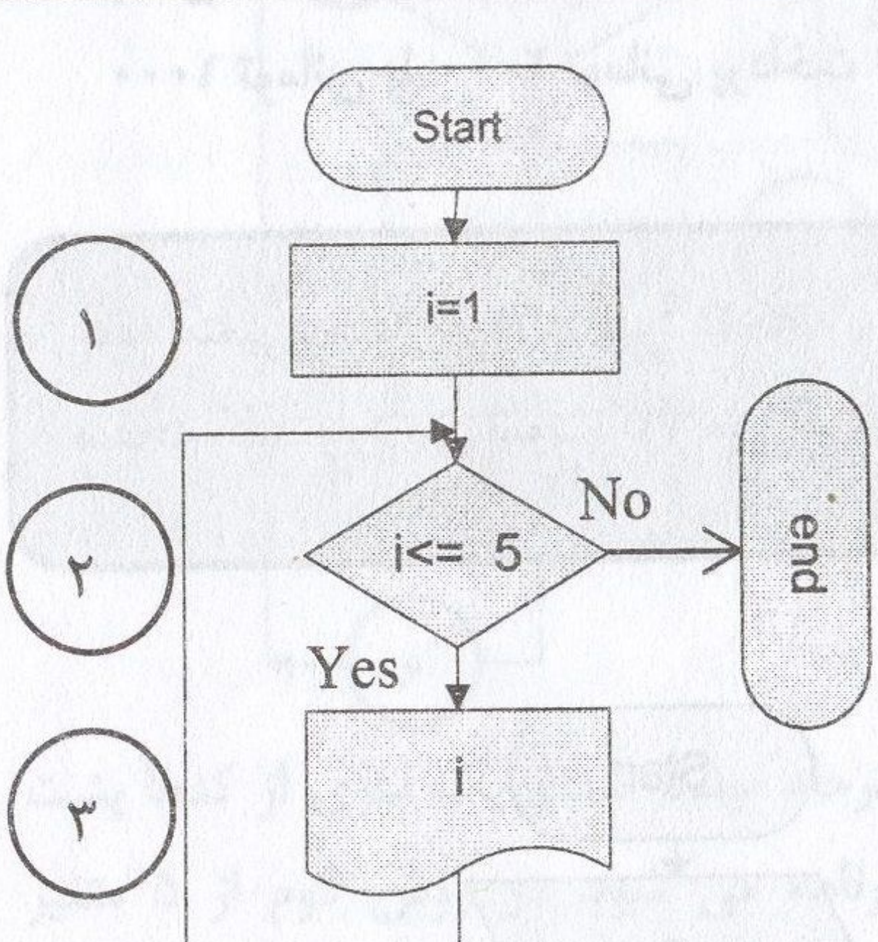
شمارنده (Counter) : حلقه باید به تعداد دفعات مشخص شمارش شود بنابراین حلقه نیاز دارد به متغیری که بتواد تعداد دفعات شمارش را در خود نگه دارد . متغیری که تعداد دفعات شمارش حلقه را در خود ذخیره می‌کند به آن **شمارنده** می گویند.

حلقه ها بطورکلی می توانند بصورت نزولی یا صعودی باشند در اشکال زیر فلوچارت هرکدام از آنها به تفکیک کشیده شده است .

حلقه نزولی : اعداد ۵ تا ۱ را بصورت نزولی چاپ می‌کند.



حلقه صعودی : اعداد ۱ تا ۵ را بصورت صعودی چاپ می‌کند.



مطابق جدول ردیابی زیر جهت چاپ اعداد ۱ تا ۵ حلقه باید ۶ بار بچرخد به ازاء هر بار چرخش یکسری نتایج مشخص می شود.

مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم	نتیجه بار ششم
1	i =1	1					
2	i <=5	yes	yes	yes	yes	yes	No
3	Print i	1	2	3	4	5	End
4	i = i +1	2	3	4	5	6	End

در مرحله یک مقدار متغیر i برابر یک می باشد و در مرحله دوم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی عدد ۵ باشد برنامه به مرحله سوم می رود و مقدار متغیر i را چاپ می کند و به مرحله چهارم می رود در این مرحله به متغیر i یک واحد اضافه می شود و برنامه به مرحله دوم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از عدد ۵ می باشد یا خیر، اگر مقدار متغیر i کوچکتر از عدد ۵ باشد این روال مطابق جدول بالا ۶ بار تکرار خواهد شد تا مقدار متغیر i از عدد ۵ بیشتر باشد تا برنامه بتواند از حلقه خارج شود .

نحوه استفاده از جدول ردیابی جهت حل الگوریتم مطابق جدول زیر توضیح داده شده است.

Step 1. Start
Step 2. X=10
Step 3. Y=5
Step 4. M=0
Step 5. M=X+Y+(X*Y)
Step 6. Y=Y+4
Step 7. M=M+Y
Step 8. Display X,Y,M
Step 9. End

Trace Table

	X	Y	M
Initial values	10	5	0
After Step 5	10	5	65
After Step 6	10	9	65
After Step 7	10	9	74

مرحله دوم : متغیر X برابر مقدار ۱۰ می باشد.
مرحله سوم : متغیر Y برابر مقدار ۵ می باشد.
مرحله چهارم : متغیر M برابر مقدار ۰ می باشد
مرحله پنجم: متغیر M برابر مقدار ۶۵ می باشد.
مرحله ششم : متغیر Y برابر مقدار ۹ می باشد.
مرحله هفتم : متغیر M برابر مقدار ۷۴ می باشد.
مرحله هشتم : متغیرهای X ، M و Y چاپ خواهند شد.
مرحله نهم : پایان برنامه می باشد.

مسئله ۵ :

برنامه ای بنویسید که اعداد زوج ۱ تا ۱۰ را کامپیوتر در خروجی چاپ کند.

۱- شروع

۲- مقدار ۲ را به متغیر i نسبت بده

۳- اگر مقدار متغیر i از عدد ۱۰ کوچکتر باشد برو به مرحله ۴

درغیراین صورت برو به

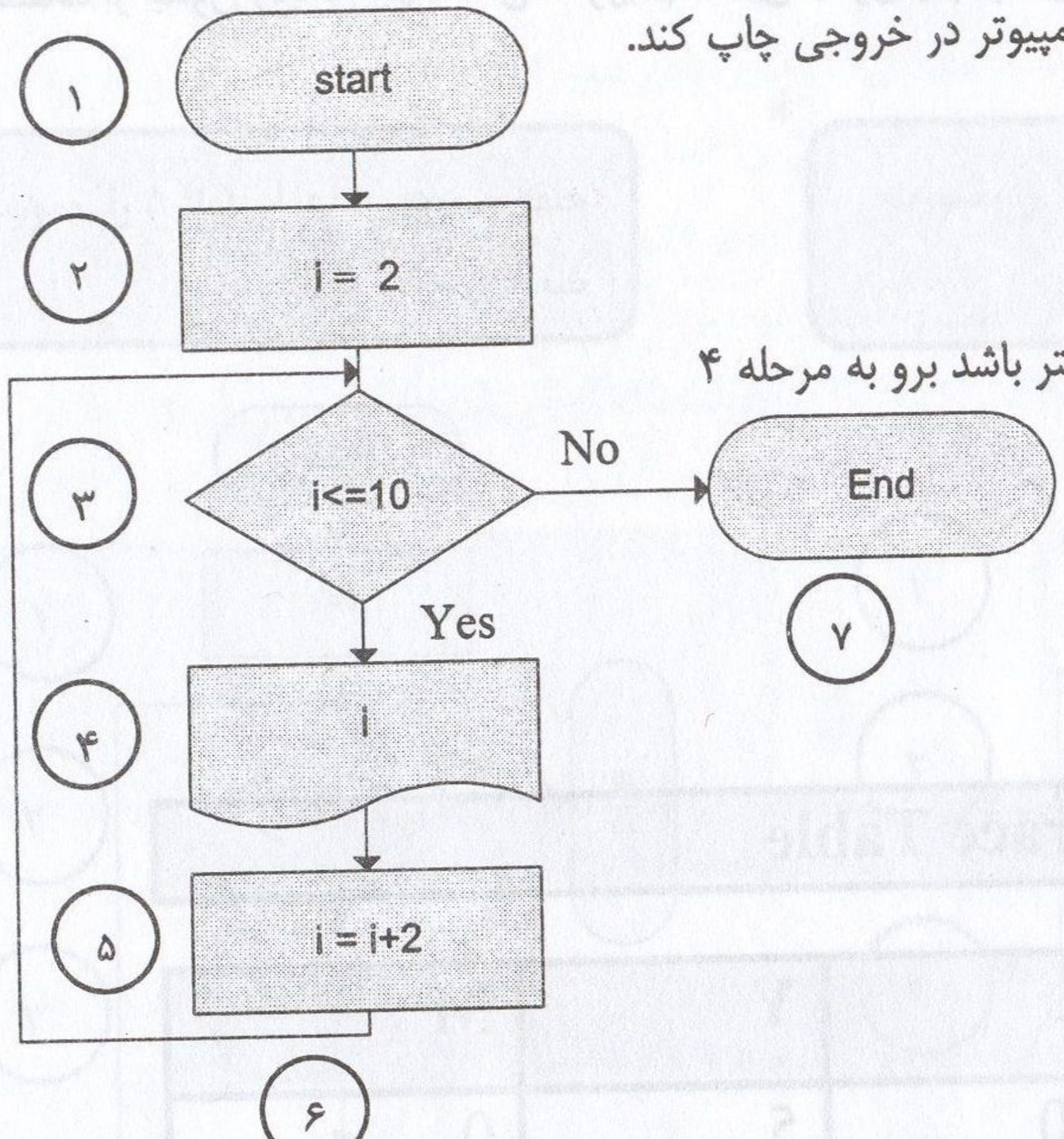
مرحله ۷

۴- مقدار متغیر i را چاپ کن

۵- به متغیر i دو واحد اضافه کن

۶- برو به مرحله ۳

۷- پایان



در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم	نتیجه بار ششم
2	i =2	2					
3	i <=10	yes	yes	yes	yes	yes	No
4	Print i	2	4	6	8	10	End
5	i = i +2	4	6	8	10	12	End

در مرحله دوم مقدار متغیر i برابر عدد ۲ می باشد و در مرحله سوم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی عدد ۱۰ باشد برنامه به مرحله چهارم می رود و مقدار متغیر i را چاپ می کند .در مرحله پنجم به متغیر i دو واحد اضافه می شود و برنامه به مرحله سوم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از عدد ۱۰ می باشد یا خیر، اگر مقدار متغیر i کوچکتر از عدد۱۰ باشد این روال مطابق جدول بالا ۶ بار تکرار خواهد شد تا مقدار متغیر i از عدد ۱۰ بیشتر باشد تا برنامه بتواند از حلقه خارج شود .

مسئله ٦ :

برنامه‌ای بنویسید کامپیوتر یک عدد از ورودی دریافت کرده از عدد ۱ تا عدد دریافت شده را در خروجی نمایش دهد.

۱- شروع

۲- متغیر n را از ورودی بخوان

۳- مقدار ۱ را به متغیر i نسبت بده

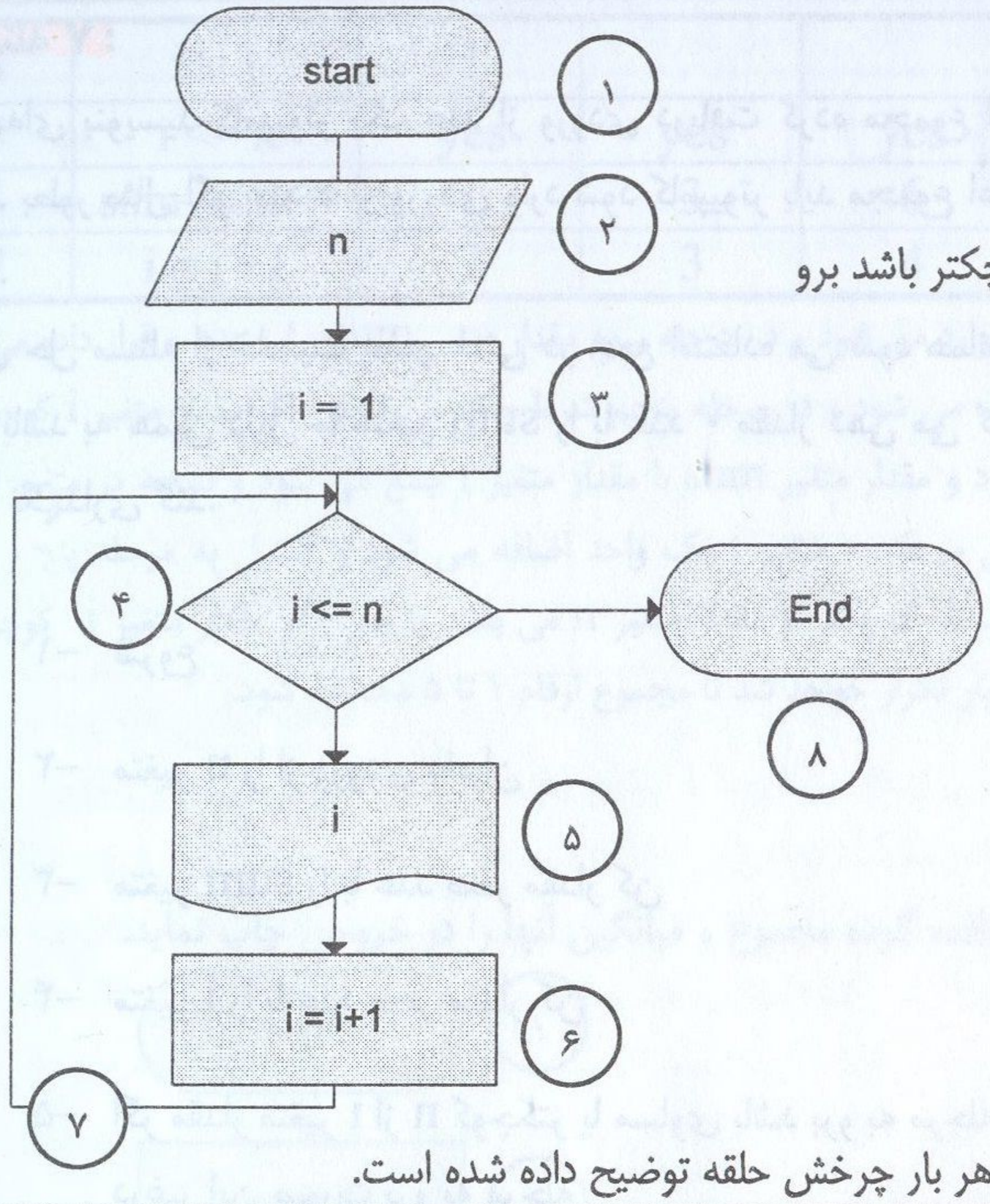
۴- اگر مقدار متغیر i از مقدارمتغیر n کوچکتر باشد برو به مرحله ۵ درغیراین صورت برو به مرحله ۸

۵- مقدار متغیر i را چاپ کن

۶- به متغیر i یک واحد اضافه کن

۷- برو به مرحله ۴

۸- پایان



در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

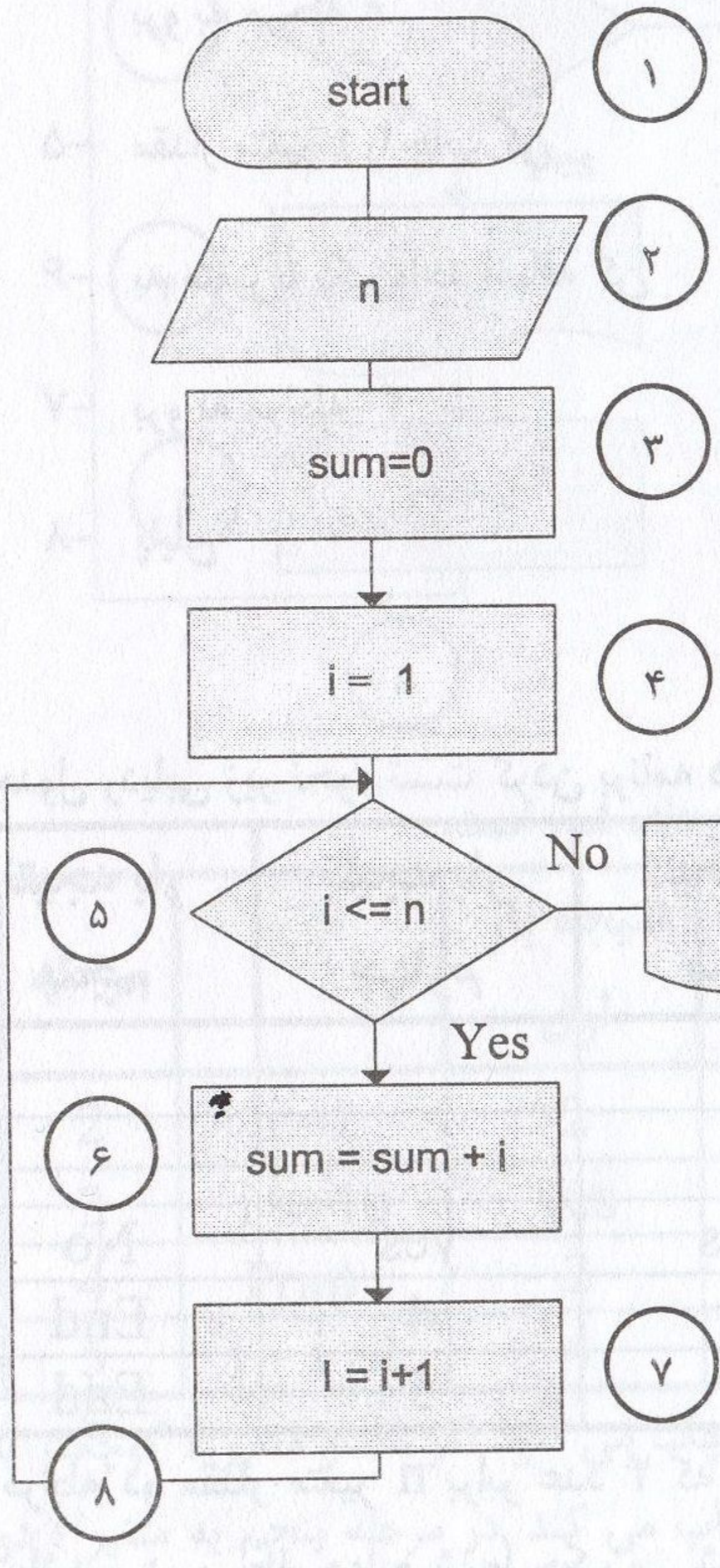
مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم
2	n	4				
3	i =1	1				
4	i <= n	yes	yes	yes	yes	No
5	Print i	1	2	3	4	End
6	i = i +1	2	3	4	5	End

در مرحله دو مقدار متغیر n برابر عدد ۴ که از ورودی دریافت می شود .در مرحله سوم مقدار متغیر i برابر عدد ۱ می باشد و در مرحله چهارم شرط چک می شود ، اگر مقدار متغیر i کوچکتر یا مساوی مقدار متغیر n باشد برنامه به مرحله پنجم می رود و مقدار متغیر i را چاپ می کند و بعد به مرحله ششم می رود در این مرحله به متغیر i یک واحد اضافه می شود و کنترل برنامه به مرحله چهارم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از n می باشد یا خیر، اگر مقدار متغیر i کوچکتر از n باشد این روال مطابق جدول بالا ۵ بار تکرار خواهد شد تا مقدار متغیر i از مقدار متغیر n بیشتر باشد تا برنامه بتواند از حلقه خارج شود .

مسئله ۷ :

برنامه‌ای بنویسید کامپیوتر یک عدد از ورودی دریافت کرده مجموع اعداد یک تا خود آن عدد را در خروجی نمایش دهد. بطور مثال اگر عدد ۵ از ورودی وارد شود کامپیوتر باید مجموع اعداد ۱ تا ۵ را محاسبه نماید و عدد ۱۵ را نمایش دهد. 1, 2,3,4,5 =>15

برای حل مسئله از خاصیت عضو خنثی در جمع استفاده می شود همانطور که می دانیم عضو خنثی در جمع عدد صفر می باشد به همین دلیل ما متغیر sum را با عدد ۰ مقدار دهی می کنیم تا در طول حلقه این متغیر مجموع ارقام در خود نگهداری کند.



۱- شروع

۲- متغیر n را از ورودی بخوان

۳- متغیر sum را با عدد صفر مقدار کن

۴- متغیر i را با عدد صفر مقدار کن

۵- اگر مقدار متغیر i از n کوچکتر یا مساوی باشد برو به مرحله
درغیر این صورت برو به مرحله

۶- مقدار متغیر i را با sum جمع کن بریز توی sum

۷- به متغیر i یک واحد اضافه کن

۸- برو به مرحله ۵

۹- مقدار متغیر sum را چاپ کن

۱۰- پایان

در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

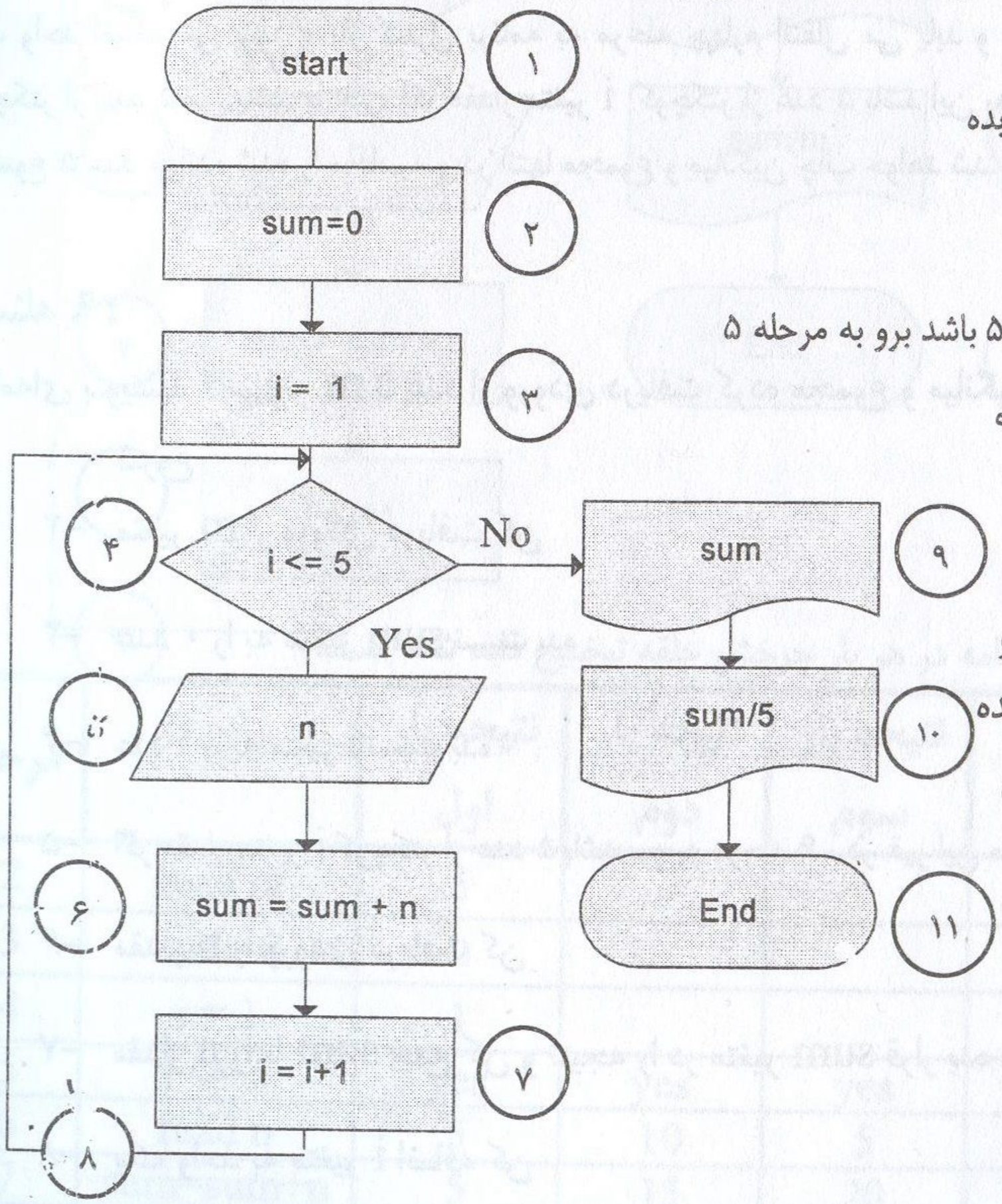
مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم	نتیجه بار ششم
2	n	5					
3	sum=0	0					

4	i = 1	1					
5	i <= n	yes	yes	yes	yes	yes	no
6	sum=sum+i	1	3	6	10	15	End
7	i = i +1	2	3	4	5	6	End

در مرحله دوم مقدار متغیر n برابر عدد ۵ از ورودی وارد می شود و در مرحله سوم مقدار متغیر sum برابر عدد ۰ قرار داده می شود ، در مرحله چهارم مقدار متغیرi برابر عدد ۱ قرار داده می شود و در مرحله پنجم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی مقدار متغیر n باشد به مرحله ششم می رود و مقدار متغیر sum با مقدار متغیر i جمع می شود و نتیجه در متغیر sum ریخته می شود و به مرحله هفتم می رود در این مرحله به متغیر i یک واحد اضافه می شود و کنترل به مرحله پنجم انتقال می‌یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از مقدار متغیر n می باشد یا خیر، اگر مقدار متغیر i کوچکتر از مقدار متغیر n باشد این روال مطابق جدول بالا ۶ بار تکرار خواهد شد تا مجموع ارقام ۱ تا ۵ محاسبه شود.

مسئله ۸:

برنامه‌ای بنویسید کامپیوتر ۵ عدد از ورودی دریافت کرده مجموع و میانگین آنها را در خروجی چاپ نماید.



۱- شروع

۲- عدد ۰ را به متغیر sum نسبت بده

۳- عدد ۱ را به متغیر i نسبت بده

۴- اگر مقدار متغیر i کوچکتر از عدد ۵ باشد برو به مرحله ۵

در غیر این صورت برو به مرحله ۹

۵- مقدار n را ورودی دریافت کن

۶- مقدار n را با sum جمع کن و نتیجه را در متغیر sum قرار بده

۷- یک واحد به متغیر i اضافه کن

۸- برو به مرحله ۴

۹- مجموع ۵ تا عدد را چاپ کن

۱۰- میانگین ۵ تا عدد را چاپ کن

۱۱- پایان

در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم	نتیجه بار ششم
2	sum=0	0					
3	i = 1	1					
4	i <= 5	yes	yes	yes	yes	yes	no
5	read n	5	10	5	20	30	
6	sum=sum+n	5	15	20	40	70	End
7	i = i +1	2	3	4	5	6	End

در مرحله دوم مقدار متغیر sum برابر عدد ۰ مقدار دهی می شود و در مرحله سوم مقدار متغیر i برابر عدد ۱ قرار داده می شود ، در مرحله چهارم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی عدد ۵ باشد برنامه به مرحله پنجم می رود در این مرحله یک عدد از ورودی می خواند و آن در متغیر n قرار می دهد و وارد مرحله ششم می شود در این مرحله مقدار متغیر sum با مقدار متغیر n جمع می شود و نتیجه در متغیر sum ریخته می شود کنترل به مرحله هفتم می رود در این مرحله به متغیر i یک واحد اضافه می شود و باز کنترل برنامه به مرحله چهارم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از عدد ۵ می باشد یا خیر، اگر مقدار متغیر i کوچکتر از عدد ۵ باشد این روال مطابق جدول بالا ۶ بار تکرار خواهد شد تا مجموع ۵ عدد خوانده شده را محاسبه و در انتها مجموع و میانگین چاپ خواهد شد. .

مسئله ۹ :

برنامه‌ای بنویسید کامپیوتر m تا عدد از ورودی دریافت کرده مجموع و میانگین آنها را در خروجی نمایش دهد.

۱- شروع

۲- متغیر m از ورودی دریافت کن

۳- عدد ۰ را به متغیر sum نسبت بده

۴- عدد ۱ را به متغیر i نسبت بده

۵- اگر مقدار متغیر i کوچکتر از عدد ۵ باشد برو به مرحله ۶ در غیر این صورت برو به مرحله ۱۰

۶- مقدار n را ورودی دریافت کن

۷- مقدار n را با sum جمع کن و نتیجه را در متغیر sum قرار بده

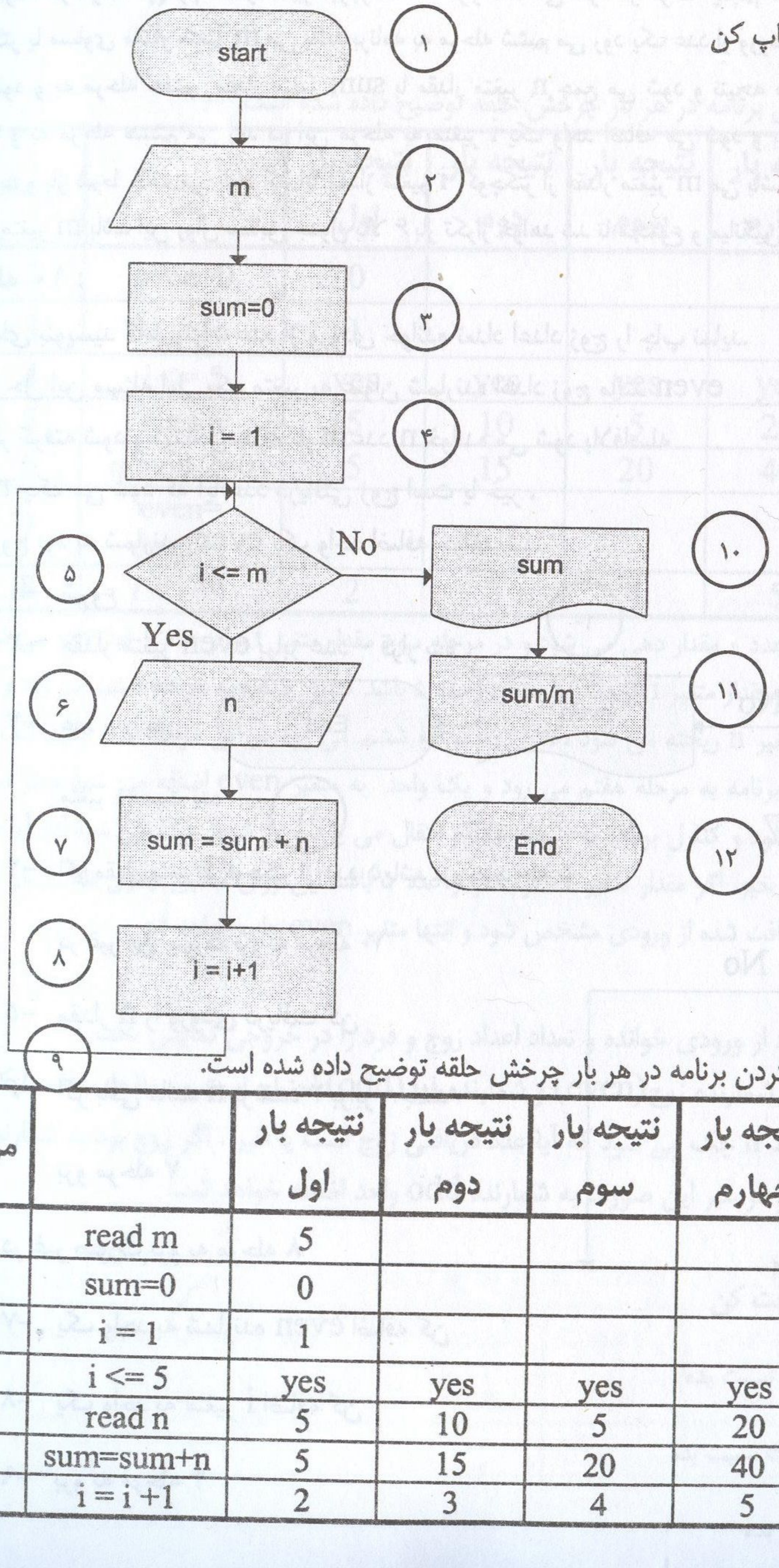
۸- یک واحد به متغیر i اضافه کن

۹- برو به مرحله ۵

۱۰- مجموع m تا عدد را چاپ کن

۱۱- میانگین m تا عدد را چاپ کن

۱۲- پایان



در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم	نتیجه بار ششم
2	read m	5					
3	sum=0	0					
4	i = 1	1					
5	i <= 5	yes	yes	yes	yes	yes	no
6	read n	5	10	5	20	30	
7	sum=sum+n	5	15	20	40	70	End
8	i = i +1	2	3	4	5	6	End

در مرحله دوم مقدار متغیر m برابر عدد ۵ از ورودی وارد می شود و در مرحله سوم مقدار متغیر sum برابر عدد ۰ قرار داده می شود ، در مرحله چهارم مقدار متغیرi برابر عدد ۱ قرار داده می شود در مرحله پنجم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی مقدار متغیر m می باشد برنامه به مرحله ششم می رود یک عدد از ورودی خوانده می شود و در متغیر n ریخته می شود و به مرحله هفتم مقدار متغیر sum با مقدار متغیر n جمع می شود و نتیجه در متغیر sum ریخته می شود کنترل برنامه و به مرحله هشتم می رود در این مرحله به متغیر i یک واحد اضافه می شود و باز کنترل برنامه به مرحله پنجم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از مقدار متغیر m می باشد یا خیر، اگر مقدار متغیر i کوچکتر از مقدار متغیر m باشد این روال مطابق جدول بالا ۶ بار تکرار خواهد شد تا مجموع و میانگین m تا عدد محاسبه شود

مسئله ۱۰ :

برنامه‌ای بنویسید کامپیوتر ۵ عدد از ورودی خوانده تعداد اعداد زوج را چاپ نماید.

جهت حل این مسئله اول یک متغیر به عنوان شمارنده اعداد زوج مانند even در نظر گرفته شود و درداخل حلقه بارکه عدد n خوانده می شود بلافاصله عدد n چک می شود که آیا عدد دریافتی زوج است یا خیر ،

اگر زوج بود به شمارنده even یک واحد اضافه خواهد شد.

۱- شروع

۲- مقدار متغیر even برابر عدد ۰ قرار بده

۳- عدد ۱ را به متغیر i نسبت بده

۴- اگر مقدار متغیر i کوچکتر از عدد ۵ باشد برو به مرحله ۵

در غیر این صورت برو به مرحله ۱۰

۵- مقدار n را ورودی دریافت کن

۶- اگر باقی مانده n بر عدد ۲ برابر ۰ باشد

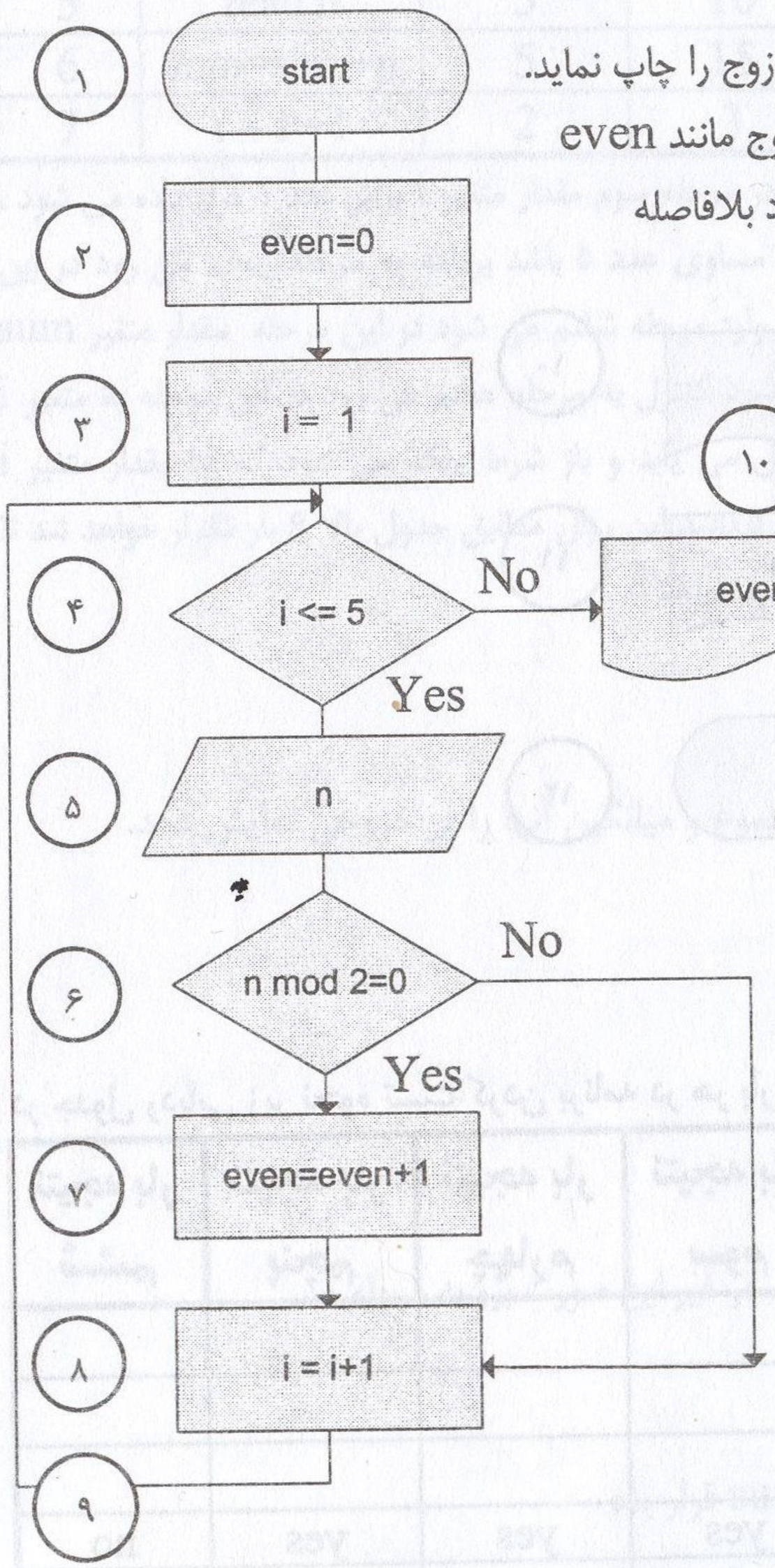
برو مرحله ۷

در غیر صورت برو به مرحله ۸

۷- یک واحد به شمارنده even اضافه کن

۸- یک واحد به متغیر i اضافه کن

۹- برو به مرحله ۴



۱۰- مقدار متغیر even را چاپ کن

۱۱- پایان

در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم	نتیجه بار ششم
2	even=0	0					
3	i = 1	1					
4	i <= 5	yes	yes	yes	yes	yes	no
5	read n	5	10	5	20	30	
6	n mod 2=0	5	15	20	40	70	End
7	even= even+1						
8	i = i +1	2	3	4	5	6	End

در مرحله دوم مقدار متغیر even برابر عدد ۰ مقدار دهی می شود و در مرحله سوم مقدار متغیر i برابر عدد ۱ قرار داده می شود در مرحله چهارم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی عدد ۵ باشد کنترل برنامه به مرحله ششم می رود و یک عدد از ورودی خوانده می شود و در متغیر n ریخته می شود ،کنترل به مرحله ششم می رود در این مرحله اگر باقیمانده مقدار متغیر n بر ۲ برابر عدد ۰ باشد کنترل برنامه به مرحله هفتم می رود و یک واحد به متغیر even اضافه می شود و در مرحله هشتم یک واحد به متغیر i اضافه می شود و کنترل برنامه به مرحله چهارم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از عدد ۵ می باشد یا خیر، اگر مقدار متغیر i کوچکتر از عدد ۵ باشد این روال مطابق جدول بالا ۶ بار تکرار خواهد شد تا تعداد اعداد زوج ۵ عدد دریافت شده از ورودی مشخص شود و انتها متغیر even چاپ خواهد شد.

مسئله ۱۱ :

برنامه‌ای بنویسید کامپیوتر m تا عدد از ورودی خوانده و تعداد اعداد زوج و فرد را در خروجی نمایش دهد.

برای حل این برنامه دو متغیر بعنوان شمارنده زوج(even) و شمارنده فرد(odd) احتیاج است و درداخل حلقه بارکه عدد n خوانده می شود بلافاصله عدد n چک می شود که آیا عدد دریافتی زوج است یا خیر ، اگر زوج بود به شمارنده even یک واحد اضافه خواهد شد و در غیر این صورت به شمارنده odd واحد اضافه خواهد شد.

۱- شروع

۲- متغیر m از ورودی دریافت کن

۳- عدد ۰ را به متغیر odd نسبت بده

۴- عدد ۰ را به متغیر even نسبت بده

۵- عدد ۱ را به متغیر i نسبت بده

۶- اگر مقدار متغیر i کوچکتر از عدد m باشد برو به مرحله ۷ در غیر این صورت برو به مرحله ۱۳

۷- مقدار n را ورودی دریافت کن

۸- اگر باقی مانده n بر عدد ۲ برابر ۰ باشد برو به مرحله ۹

در غیر این صورت برو به مرحله ۱۰

۹- یک واحد به متغیر odd اضافه کن برو به مرحله ۱۱

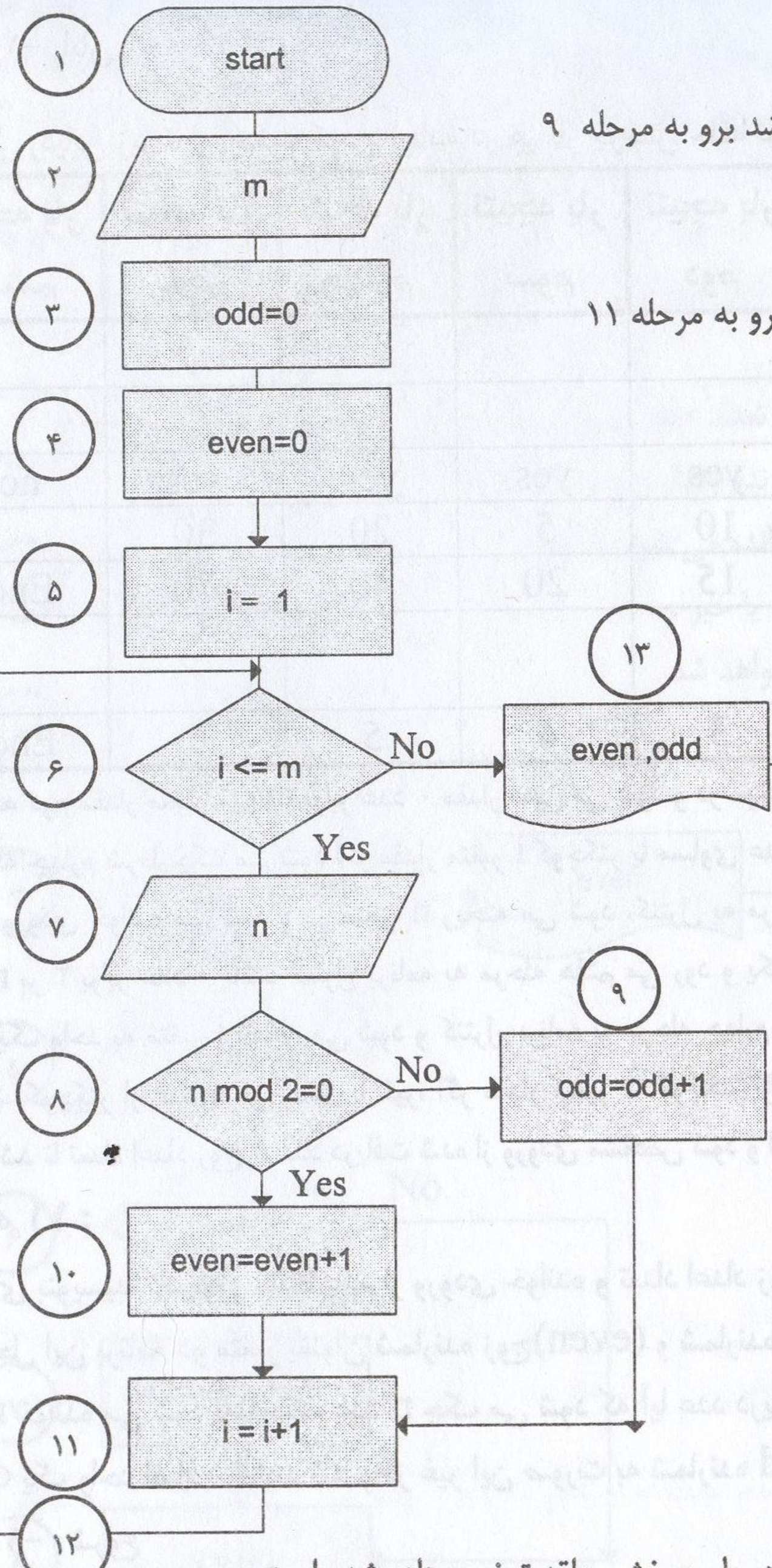
۱۰- یک واحد به متغیر even اضافه کن

۱۱- یک واحد به متغیر i اضافه کن

۱۲- برو به مرحله ۶

۱۳- مقادیر متغیر

۱۴- پایان.



های even و odd را چاپ کن

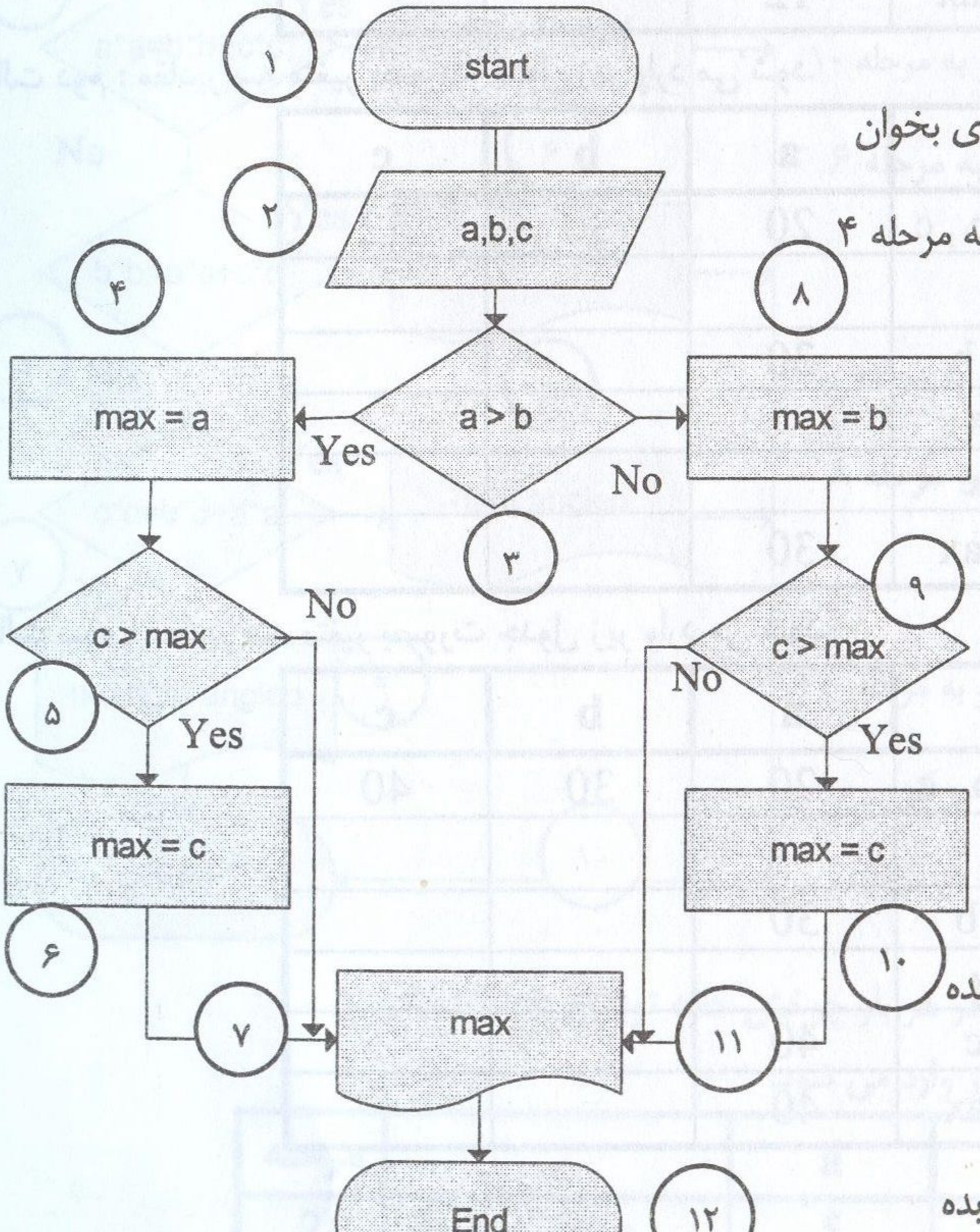
در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم	نتیجه بار ششم
2	read m	5					
3	odd=0						

4	even=0	0					
5	i = 1	1					
6	i <= 5	yes	yes	yes	yes	yes	no
7	read n	5	10	51	20	30	
8	n mod 2=0	no	yes	no	yes	yes	End
9	odd = odd+1	1		2			
10	even = even+1		1		2	3	
11	i = i +1	2	3	4	5	6	End

مسئله ۱۲ :

برنامه ای بنویسید کامپیوتر اضلاع یک مثلث را بعنوان ورودی بگیرد و بگوید کدام یک از همه بزرگتر می باشد.



۱- شروع

۲- سه متغیر بنامهای a ,b,c از ورودی بخوان

۳- اگر متغیر a بزرگتر از b باشد برو به مرحله ۴

درغیر این صورت برو به مرحله

۴- متغیر a را به متغیر max

نسبت بده

۵- اگر متغیر c بزرگتر ازمتغیر

max باشد برو مرحله ۶

در غیر این صورت برو به مرحله ۷

۶- متغیر c را به متغیر max نسبت بده

۷- max را چاپ کن

۸- متغیر b را به متغیر max نسبت بده

۹- اگر متغیر c بزرگتر ازمتغیر max باشد برو مرحله ۱۰ غیر این صورت برو به مرحله ۱۱

۱۰- متغیر c را به متغیر max نسبت بده

۱۱- max را چاپ کن

۱۲- پایان

در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

حالت اول : مقادیر سه متغیر بصورت جدول زیر وارد می شود.

مرحله		a	b	c
2	read a , b , c	10	7	12
3	a > b			
4	max = a	10		
5	c>max			
6	max= c			
7	print max	12		

حالت دوم : مقادیر سه متغیر بصورت جدول زیر وارد می شود.

مرحله		a	b	c
2	read a , b , c	20	30	12
3	a > b			
8	max = b	30		
9	c>max			
10	max= c			
11	print max	30		

حالت سوم : مقادیر سه متغیر بصورت جدول زیر وارد می شود.

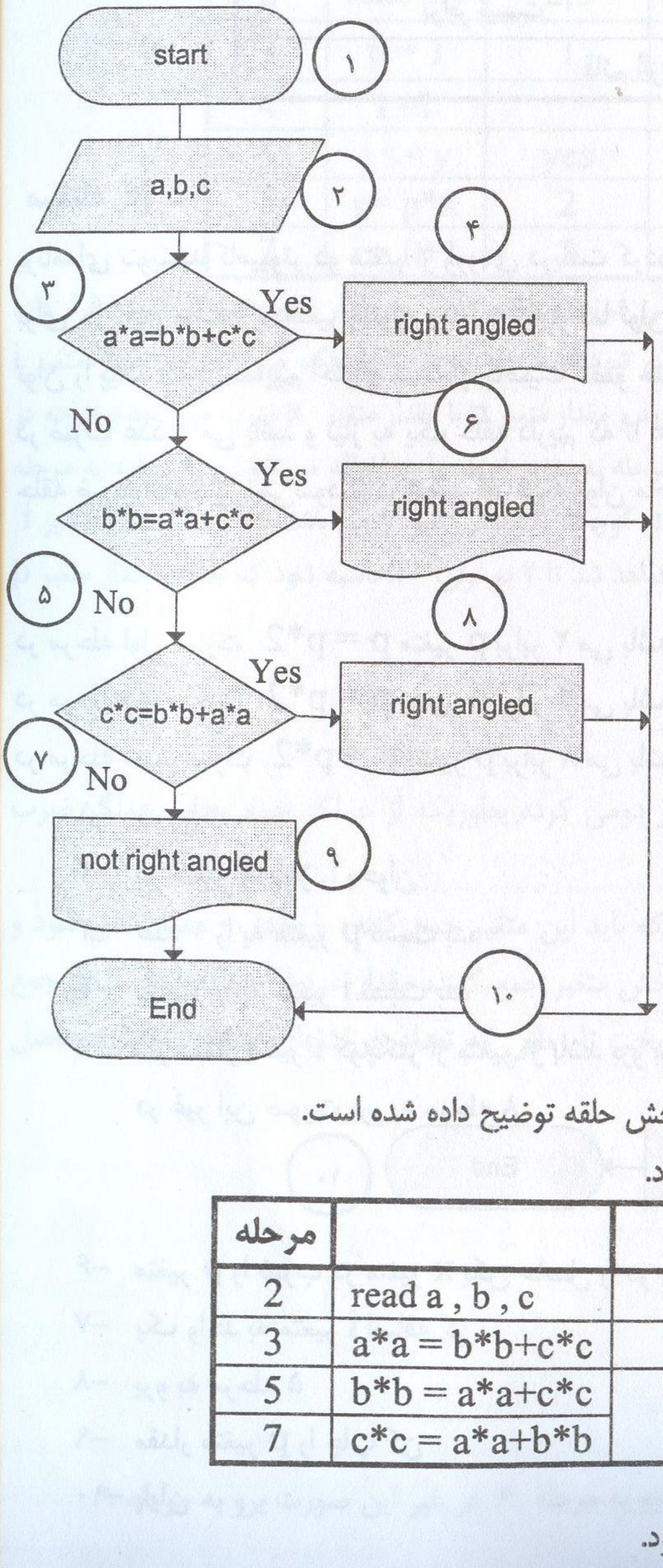
مرحله		a	b	c
2	read a , b , c	20	30	40
3	a > b			
8	max = b	30		
9	c>max			
10	max= c	40		
11	print max	40		

مسئله ۱۳ :

برنامه‌ای بنویسید کامپیوتر اضلاع یک مثلث را از ورودی خوانده اگر مثلث قائم الزاویه باشد یک پیغام در خروجی نمایش دهد که مثلث قائم الزاویه می باشد .

جهت حل این مسئله از قیضه قیثاغورث استفاده می شود در این قضیه زمانی مثلث می تواند قائم الزاویه باشد که باید وتر به توان دو برابر با مجموع هر کدام از اضلاع به توان باشد و ممکن است هر کدام از اضلاع بعنوان وتر در نظر گرفته شود . لذا باید سه حالت زیر را چک کرد تا برنامه درست جواب دهد.

a*a = b*b+c*c , b*b = a*a+c*c , c*c = a*a+b*b



۱- شروع

۲- سه متغیر a,b,c را از ورودی بخوان

۳- اگر a*a = b*b+c*c باشد برو به مرحله ۴

در غیر این صورت برو به مرحله ۵

۴- پیغام مثلث قائم الزاویه را چاپ کن برو به مرحله ۱۰

۵- اگر b*b = a*a+c*c باشد بروبه مرحله ۶

در غیراین صورت برو به مرحله ۷

۶- پیغام مثلث قائم الزاویه را چاپ کن برو به مرحله ۱۰

۷- اگر c*c = a*a+b*b باشد بروبه مرحله ۸

در غیراین صورت برو به مرحله ۹

۸- پیغام مثلث قائم الزاویه را چاپ کن برو به مرحله ۱۰

۹- پیغام مثلث قائم الزاویه نیست را چاپ کن

۱۰- پایان

در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

حالت اول : مقادیر سه متغیر بصورت جدول زیر وارد می شود.

مرحله		a	b	c
2	read a , b , c	5	3	4
3	a*a = b*b+c*c	5*5=3*3+4*4		
5	b*b = a*a+c*c	قائم الزاویه است		
7	c*c = a*a+b*b			

حالت دوم : مقادیر سه متغیر بصورت جدول زیر وارد می شود.

مرحله		a	b	c
2	read a , b , c	7	3	4
3	a*a = b*b+c*c	برابر نیست		
5	b*b = a*a+c*c	برابر نیست		
7	c*c = a*a+b*b	برابر نیست		
قائم الزاویه نیست				

مسئله ۱۴ :

برنامه‌ای بنویسید کامپیوتر دو متغیر از ورودی دریافت کرده و اولی را به توان دومی برساند.

برای حل این مسئله دو متغیر بعنوان بعنوان پایه و نما توان مانند x و y درنظر گرفته می شود . حال اگر بخواهیم عمل توان را پیاده سازی نماییم احتیاج است از خاصیت عضو خنثی در ضرب استفاده شود . همانطور که می دانید عضو خنثی در ضرب عدد ۱ می باشد و نیاز به یک حلقه داریم که تا عدد دوم باید شمارش شود و مقدار متغیر عضو خنثی در داخل حلقه ضربدر عدد اول می‌شود تا در اتمام کار حلقه توان محاسبه شود.

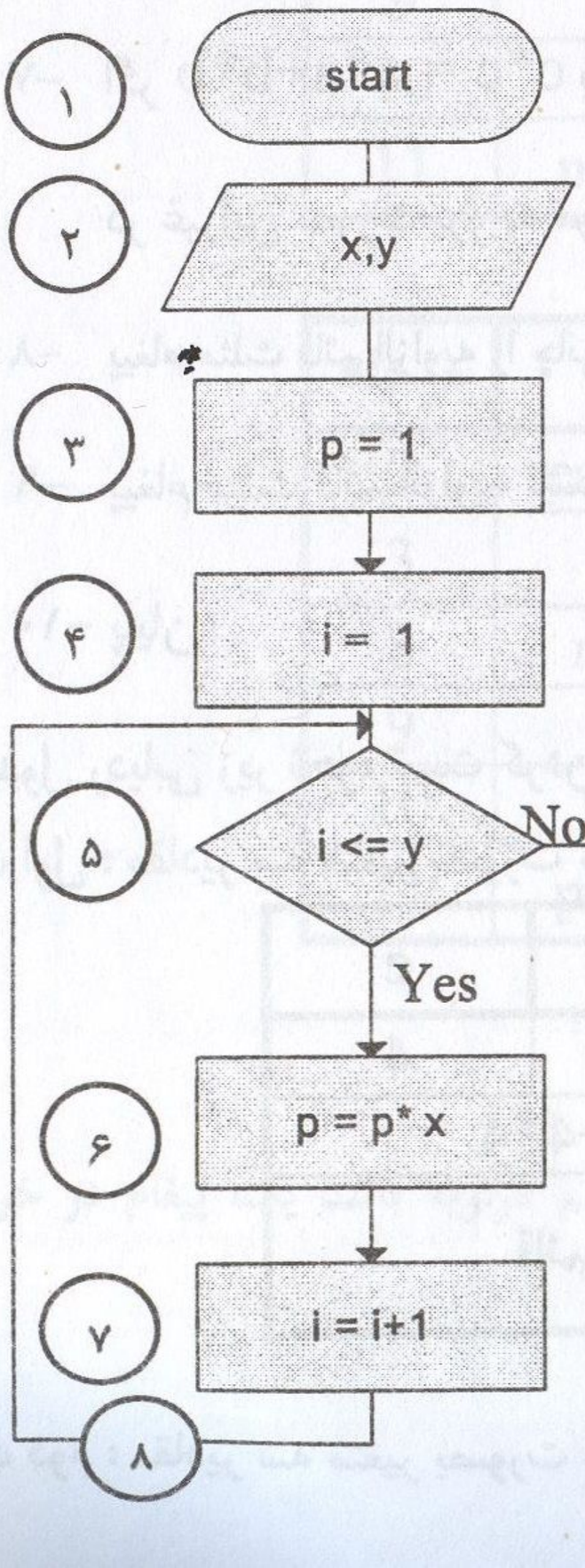
p=1; ① ② ③

2^3 =2*2*2 , p = p*2 , p = p*2, p= p*2

در مرحله اول عبارت p = p*2 متغیر p برابر ۲ می باشد.

در مرحله دوم عبارت p = p*2 متغیر p برابر ۴ می باشد.

در مرحله سوم عبارت p = p*2 متغیر p برابر ۸ می باشد.



۱- شروع

۲- دو متغیر x و y را بخوان

۳- عدد ۱ را به متغیر p نسبت بده

۴- عدد ۱ را به متغیر i نسبت بده

۵- اگر مقدار متغیر i کوچکتر از متغیر y باشد برو به مرحله ۶

در غیر این صورت برو به مرحله ۸

۶- متغیر p را ضرب در متغیر x بکن حاصل را در متغیر p بریز

۷- یک واحد به متغیر i اضافه کن

۸- برو به مرحله ۵

۹- مقدار متغیر p را چاپ کن

۱۰- پایان

در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم
2	read x,y	2,3			
3	p = 1	1			
4	i =1				
5	i <= y	yes	yes	yes	no
6	p= p*x	2	4	8	end
7	i = i +1	2	3	4	end

در مرحله دوم دو متغیر x و y از ورودی به ترتیب ۲و ۳ خوانده می شوند و در مرحله سوم مقدار متغیر p برابر عدد ۱ قرار داده می شود ، در مرحله چهارم مقدار متغیرi برابر عدد ۱ قرار داده می شود در مرحله پنجم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی مقدار متغیر y باشد برنامه به مرحله ششم می رود و مقدار متغیر p با مقدار متغیر x ضرب می شود و نتیجه در متغیر p ریخته می شود برنامه و به مرحله هفتم می رود در این مرحله به متغیر i یک واحد اضافه می شود و باز برنامه به مرحله پنجم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از مقدار متغیر y می باشد یا خیر، اگر مقدار متغیر i کوچکتر از عدد y باشد این روال مطابق جدول بالا ۴ بار تکرار خواهد شد تا ۲ به توان ۳ محاسبه شود که حاصل مقدار متغیر p برابر ۸ خواهد شد.

مسئله ۱۵ :

برنامه‌ای بنویسید دو عدد از ورودی خوانده اولی را ضرب در دومی کرده بطوریکه از عملگر جمع بجای عملگر ضرب استفاده نمایید.

برای حل این مسئله احتیاج به یک متغیر جمع کننده داریم که باید این متغیر جمع کننده با مقدار ۰ مقدار دهی شود و سپس به حلقه که به تعداد عدد دوم باید بچرخد و هر بار چرخش متغیر جمع کننده مقدار عدد اول را با مقدار متغیر جمع کننده ، جمع کرده و نتیجه را در متغیر جمع کننده خواهد ریخت تا زمانی که تعداد چرخشهای حلقه تمام شود حاصل ضرب با عملگر جمع شبیه سازی خواهد شد.

۱- شروع

۲- دو متغیر x و y را بخوان

۳- عدد ۰ را به متغیر sum نسبت بده

۴- عدد ۱ را به متغیر i نسبت بده

۵- اگر مقدار متغیر i کوچکتر از مقدار متغیر y باشد برو به مرحله ۶ در غیر این صورت برو به مرحله ۹

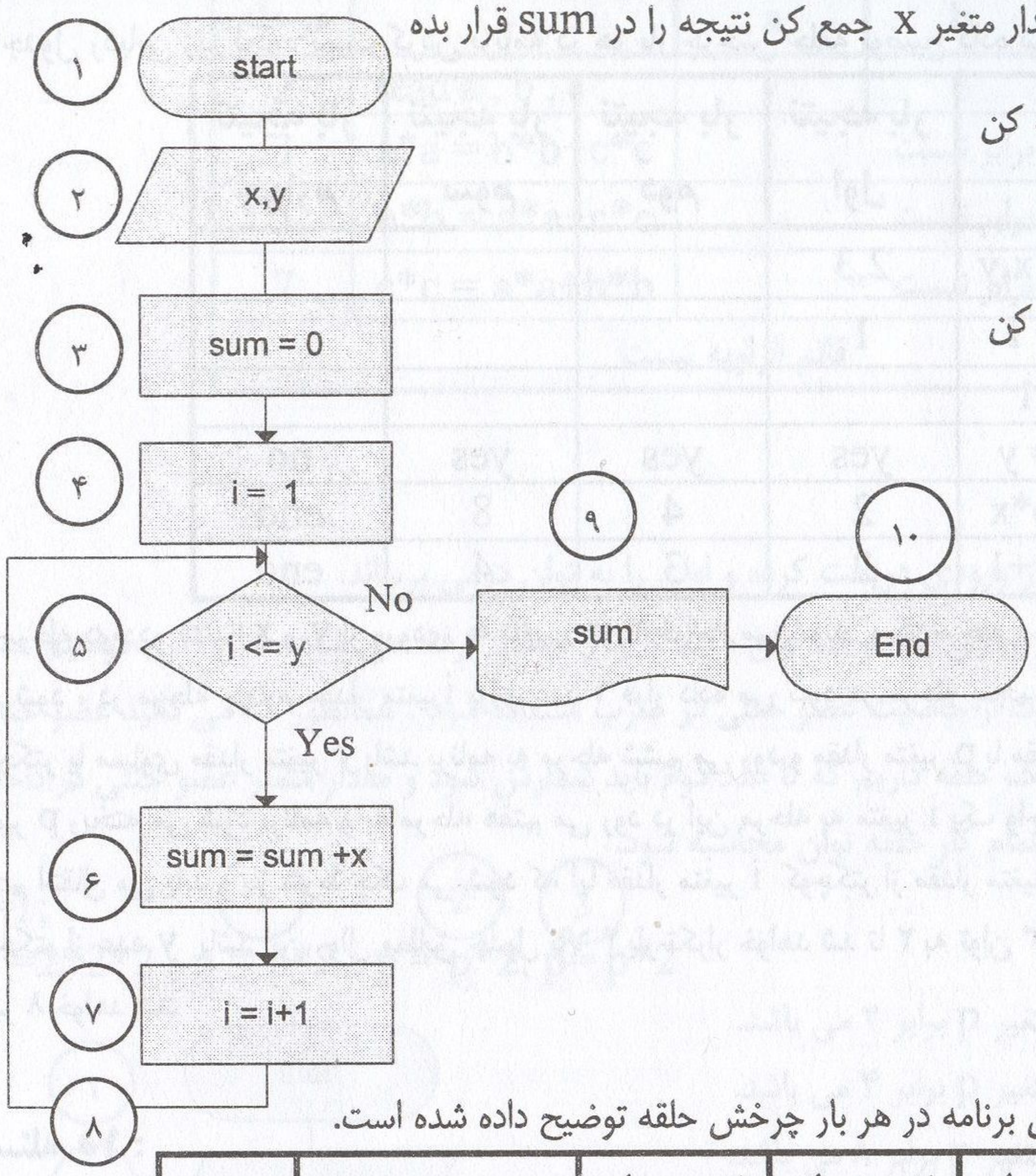
۶- مقدار متغیر sum را با مقدار متغیر x جمع کن نتیجه را در sum قرار بده

۷- به متغیر i یک واحد اضافه کن

۸- برو به مرحله ۵

۹- مقدار متغیر sum را چاپ کن

۱۰- پایان



در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

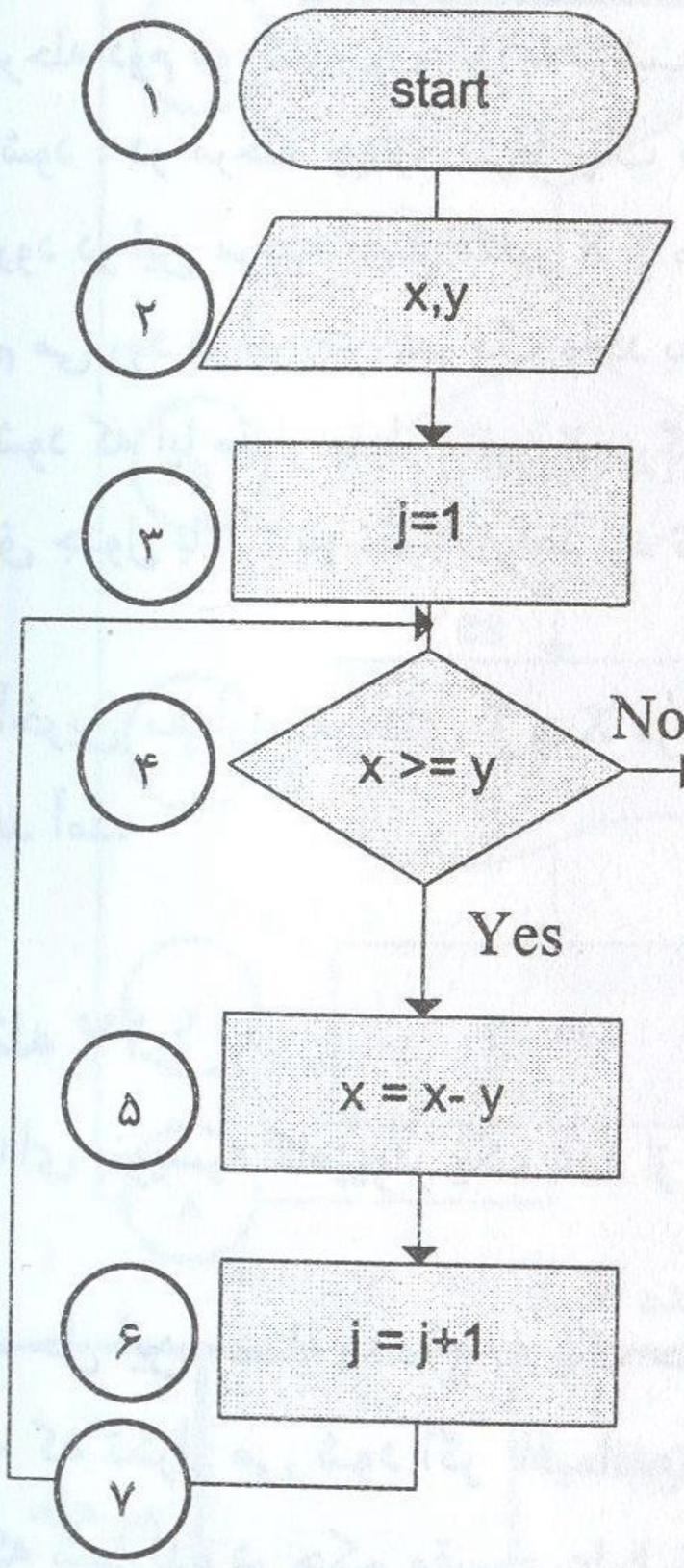
مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم
2	read x,y	2,3			
3	sum =0	0			
4	i =1				
5	i <= y	yes	yes	yes	no
6	sum = sum+x	2	4	6	end
7	i = i +1	2	3	4	end

در مرحله دوم دو متغیر x و y از ورودی به ترتیب ۲و ۳ خوانده می شوند و در مرحله سوم مقدار متغیر sum برابر عدد ۰ قرار داده می شود ، در مرحله چهارم مقدار متغیرi برابر عدد ۱ قرار داده می شود در مرحله پنجم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی مقدار متغیر y باشد برنامه به مرحله ششم می رود و مقدار متغیر sum با مقدار متغیر x جمع می شود و نتیجه در متغیر sum ریخته می شود کنترل به مرحله هفتم می رود در این مرحله به متغیر i یک واحد اضافه می شود و باز کنترل برنامه به مرحله پنجم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از مقدار متغیر y می باشد یا خیر، اگر مقدار متغیر i کوچکتر از عدد y باشد این روال مطابق جدول بالا ۴ بار تکرار خواهد شد تا ۲ ضرب در ۳ محاسبه شود که حاصل مقدار متغیر p برابر عدد ۶ خواهد شد.

مسئله ۱۶ :

برنامه‌ای بنویسید دو عدد از ورودی خوانده اولی را تقسیم بر دومی کرده بطوریکه از عملگر تفریق بجای عملگر تقسیم استفاده نمایید و همچنین خارج قسمت و باقیمانده را کامپیوتر در خروجی نمایش دهد. با این فرض که عدد اول بزرگتر از عدد دوم می باشد.

جهت حل این مسئله باید از طریق حلقه عدد اول را از عدد دوم آنقدر کم کنیم تا زمانی که عدد اول کوچکتر از عدد دوم باشد به ازاء هر بار کم کردن یک واحد به شمارنده اضافه شود. شمارنده در حکم خارج قسمت خواهد بود و آخرین مقداری که عدد اول از عدد دوم کم می شود در حکم باقیمانده خواهد بود.



۱- شروع

۲- دو متغیر x و y را از ورودی دریافت کن

۳- عدد ۱را به متغیر j نسبت بده این متغیر بعنوان خارج قسمت می باشد.

۴- اگر مقدار متغیر x بزرگتر y باشد برو به مرحله در غیر این صورت برو به مرحله

۵- مقدارمتغیر x از مقدار متغیر متغیر y کم کن و نتیجه را در متغیر x قرار بده

۶- یک واحد به متغیر j اضافه کن

۷- برو به مرحله ۴

۸- مقدار j را چاپ کن و مقدار x را چاپ کن

۹- پایان

نکته۱ : در این برنامه خارج قسمت در j ذخیره خواهد شد و باقیمانده در متغیر x ذخیره خواهد . توسط حلقه متغیر x بعد از تفریقهای مکرر به جای خواهد رسید که از متغیر y کوچکتر خواهد بود در این زمان حلقه به خود پایان خواهد داد

نکته ۲ : این مدل حلقه متفاوت با حلقه های قبلی می باشد بطوریکه در حلقه های قبلی برای این حلقه به خود پایان دهد یک شمارنده مخصوص حلقه بود و تعداد دفعات مشخص بود ولی در این نوع حلقه تعداد دفعات مشخص نیست و بستگی به دو عدد دریافتی از صفحه کلید دارد.

در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم
2	read x,y	10 , 3			
3	j = 0	0			
4	x > y	yes	yes	yes	no
5	x = x-y	7	4	1	end
6	j = j +1	1	2	3	end

در مرحله دوم دو متغیر x و y به ترتیب ۱۰ و ۳ از ورودی دریافت می شود و در مرحله سوم مقدار متغیر j برابر عدد ۰ قرار داده می شود ، در مرحله چهارم شرط چک می شود اگر مقدار متغیر x بزرگتر از مقدار متغیر y باشد کنترل برنامه به مرحله پنجم می رود در این مرحله مقدار متغیر x از مقدار متغیر y کم می شود و نتیجه در متغیر x ذخیره می شود و کنترل برنامه به مرحله ششم می رود در این مرحله یک واحد به متغیر j اضافه می شود و کنترل برنامه به مرحله چهارم انتقال می یابد و باز شرط چک می شود که آیا مقدار مقدار متغیر x بزرگتر از مقدار متغیر y می باشد یا خیر، اگر متغیر x بزرگتر از مقدار متغیر y باشد این روال مطابق جدول بالا ۴ بار تکرار خواهد شد تا متغیر j خارج قسمت و متغیر x باقیمانده را نمایش دهد و برای تست این عمل

n=j*y+x => n=3*3+1

اگر آخرین مقدار متغیرهای j و x در عبارت بالا جایگزین شود مقدار عدد مقسوم که همان عدد ۱۰ می باشد بدست خواهد آمد.

مسئله ۱۷ :

برنامه‌ای بنویسید کامپیوتر یک عدد از ورودی دریافت کرده مقسوم علیه های آن را در خروجی نمایش دهد.

12 ==> 1, 2 , 3 , 4, 6 , 12

جهت حل این مسئله احتیاج به یک حلقه می باشد این حلقه به تعداد دفعات عدد دریافتی ورودی باید تکرار شود در هر مرتبه که تکرار می شود اگر باقیمانده عدد دریافتی از ورودی بر شمارنده حلقه برابر عدد ۰ باشد شمارنده چاپ خواهد شد که شمارنده در حکم مقسوم علیه خواهد بود .

عدد دریافتی	12	12	12	12	12	12	12	12	12	12	12	12
if(n mod 2)=0	yes	yes	yes	yes	no	yes	no	no	no	no	no	yes
شمارنده	1	2	3	4	5	6	7	8	9	10	11	12
چاپ	1	2	3	4		6						12

۱- شروع

۲- متغیر n را از ورودی بخوان

۳- عدد ۱ را به متغیر i نسبت بده

۴- اگر مقدار متغیر i کوچکتر از متغیر n باشد برو به مرحله ۵

در غیر این صورت برو به مرحله ۹

۵- اگر باقیمانده متغیر n بر مقدار متغیر i برابر عدد ۰ باشد

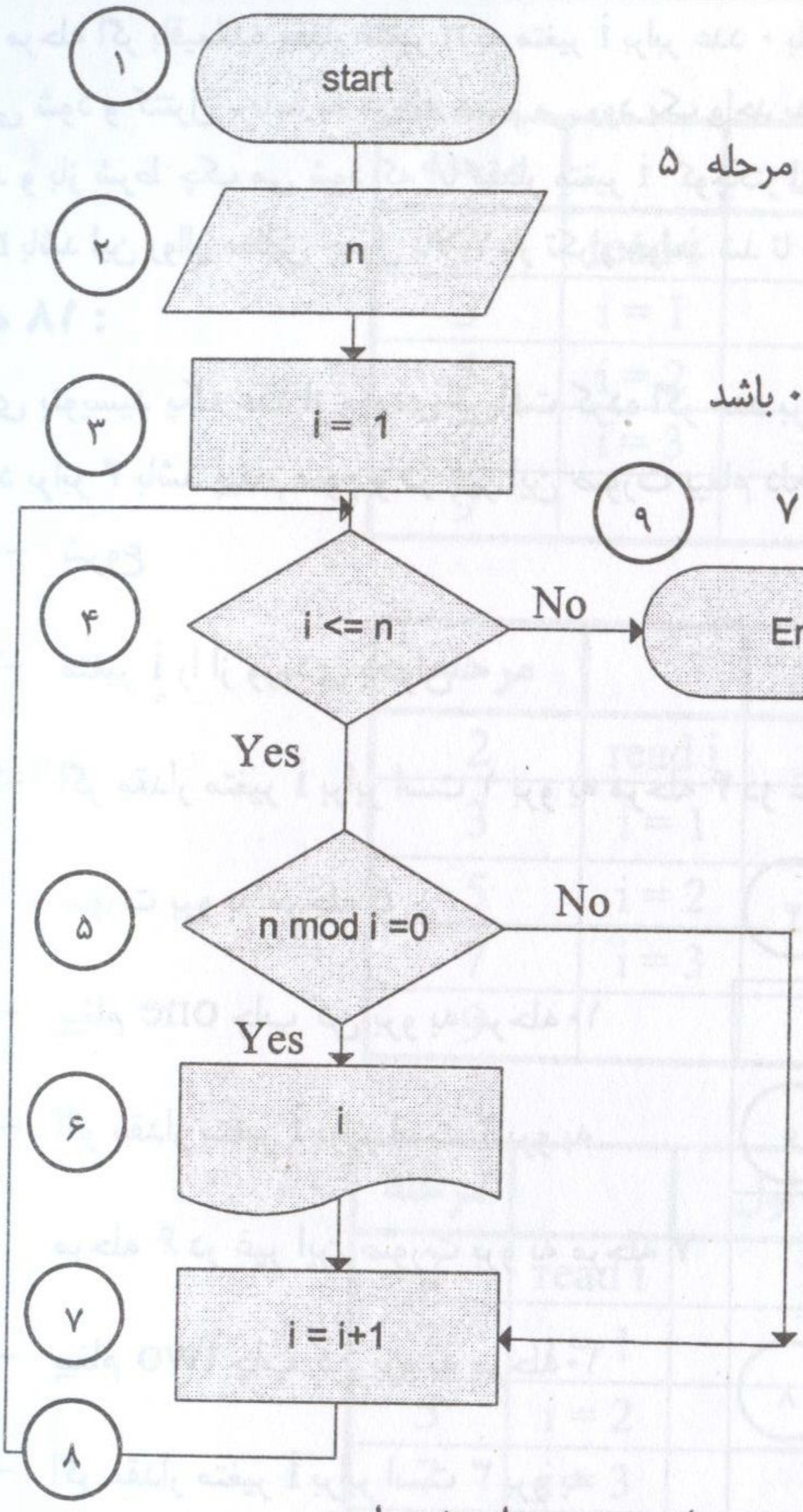
برو به مرحله ۶ در غیر این صورت برو به مرحله ۷

۶- مقدار متغیر i را چاپ کن

۷- یک واحد به متغیر i اضافه کن

۸- برو به مرحله ۴

۹- پایان.



در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

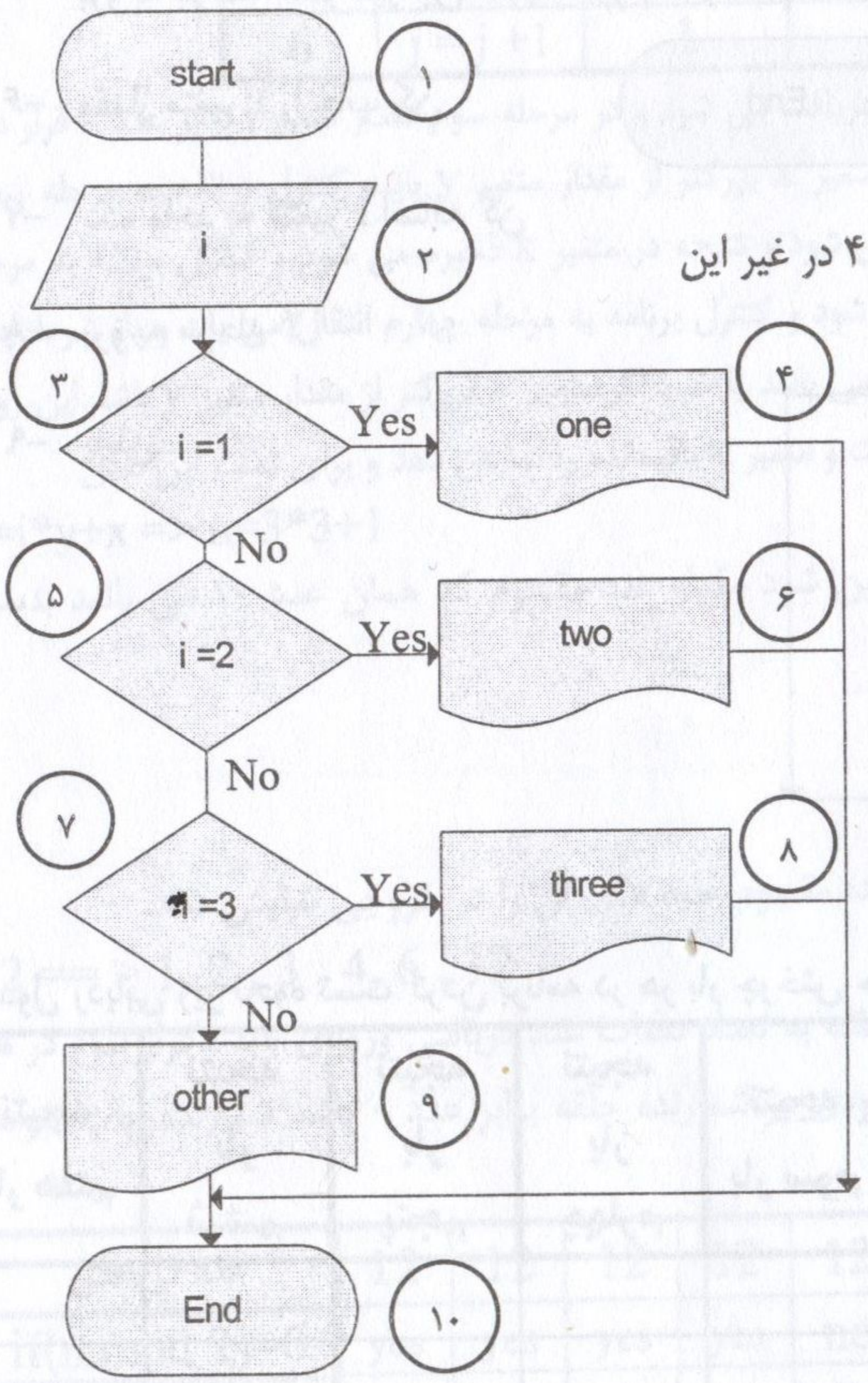
مرحله		نتیجه بار اول	نتیجه بار دوم	نتیجه بار سوم	نتیجه بار چهارم	نتیجه بار پنجم	نتیجه بار ششم	نتیجه بار هفتم
2	read n	6						
3	i = 1	1						
4	i <= n	yes	yes	yes	yes	yes	yes	no
5	n mod i=0	yes	yes	yes	no	yes	yes	end
6	print i	1	2	3			6	end
7	i = i +1	2	3	4	5	6	7	end

در مرحله دوم مقدار متغیر n از ورودی عدد ۶ وارد می شود و در مرحله سوم مقدار متغیر i برابر عدد ۱ قرار داده می شود ، د
مرحله چهارم شرط چک می شود اگر مقدار متغیر i کوچکتر یا مساوی مقدار متغیر n باشد کنترل برنامه به مرحله پنجم می رو

در این مرحله اگر باقیمانده مقدار متغیر n بر متغیر i برابر عدد ۰ باشد کنترل به مرحله ششم می رود در این مرحله مقدار متغیر i چاپ می شود و کنترل برنامه به مرحله هفتم می رود یک واحد به متغیر i اضافه می شود و کنترل برنامه به مرحله چهارم انتقال می یابد و باز شرط چک می شود که آیا مقدار متغیر i کوچکتر از مقدار n می باشد یا خیر، اگر مقدار متغیر i کوچکتر از مقدار متغیر n باشد این روال مطابق جدول بالا ۷ بار تکرار خواهد شد تا مقسوم علیه های عدد ۶ را کامپیوتر چاپ کند.

مسئله ۱۸ :

برنامه‌ای بنویسید یک عدد از ورودی دریافت کرده اگر عدد برابر ۱ باشد پیغام اول و اگر عدد برابر ۲ باشد پیغام دوم و اگر عدد برابر ۳ باشد پیغام سوم و در غیر این صورت پیغام دلخواه چاپ شود.



۱- شروع

۲- متغیر i را از ورودی بخوان

۳- اگر مقدار متغیر i برابر است ۱ برو به مرحله ۴ در غیر این صورت برو به مرحله ۵

۴- پیغام one چاپ کن برو به مرحله۱۰

۵- اگر مقدار متغیر i برابر است ۲ برو به مرحله ۶ در غیر این صورت برو به مرحله ۷

۶- پیغام two چاپ کن برو به مرحله۱۰

۷- اگر مقدار متغیر i برابر است ۳ برو به مرحله ۸ در غیر این صورت برو به مرحله ۹

۸- پیغام three چاپ کن برو به مرحله ۱۰

۹- پیغام other چاپ کن برو به مرحله ۱۰

۱۰- پایان

در جدول ردیابی زیر نحوه تست کردن برنامه در هر بار چرخش حلقه توضیح داده شده است.

حالت اول : اگر مقدار متغیر i برابر عدد ۱ وارد می شود.

مرحله		نتیجه بار اول	
2	read i	1	
3	i = 1	yes	one
5	i = 2		
7	i = 3		
9			

حالت دوم : اگر مقدار متغیر i برابر عدد ۲ وارد می شود.

مرحله		نتیجه بار اول	
2	read i	2	
3	i = 1	no	
5	i = 2	yes	two
7	i = 3		
9			

حالت سوم : مقدار متغیر i برابر عدد ۳ وارد می شود.

مرحله		نتیجه بار اول	
2	read i	3	
3	i = 1	no	
5	i = 2	no	
7	i = 3	yes	three
9			

حالت چهارم : اگر مقدار متغیر i برابر عدد ۵ وارد می شود.

مرحله		نتیجه بار اول	
2	read i	1	
3	i = 1	no	
5	i = 2	no	
7	i = 3	no	
9			other