# مبانی کامپیوتر و برنامه‌نویسی به زبان C

# فصل دوم: مراحل انجام یک کار با استفاده از کامپیوتر

## ۲-۱ مقدمه

**تفاوتهای عمده انسان و کامپیوتر**

## ۲-۲ مراحل حل یک مسئله با استفاده از کامپیوتر

**۱- تعریف صورت مسئله به طور جامع و دقیق.**

**۲- تعیین راه حل قدم به قدم (الگوریتم) برای حل مسئله.**

**۳- تبدیل راه حل قدم به قدم یا الگوریتم به روند نما.**

**۴- بررسی درستی روندنما با استفاده از داده‌های آزمایشی.**

**۵- نوشتن برنامه با استفاده از یک زبان برنامه‌نویسی.**

**۶- واردکردن برنامه به کامپیوتر و اشکال‌زدایی آن.**

**۷- آزمایش برنامه با استفاده از داده‌های آزمایشی.**

**۸- استفاده از برنامه.**

## ۲-۲-۱ تعریف صورت مسئله به طور جامع و دقیق

**مسئله جمع زدن تعدادی عدد**

*تعدادی عدد صحیح و مثبت در دست است و پایان آن‌ها با یک عدد صفر مشخص شده است. برنامه‌ای بنویسید که این عددها را از ورودی خوانده و پس از چاپ آن عددها، حاصل جمع آن‌ها را نیز چاپ نماید.*

**نمی‌توان ادعا کرد که این تعریف مسئله یا هر تعریف دیگری صد در صد خالی از ابهام است. هزینه ای بین ۱۰۰ تا ۲۰۰ برابر برای برطرف کردن اشکال صورت مسئله پس از خاتمه کار.**

## ۲-۲-۲ تعیین راه حل قدم به قدم (الگوریتم)

**وجود یک مسئله.**

- **نیاز به یک راه حل از طریق تفکر توسط طراح الگوریتم (راه حل)، مهمترین مرحله.**
- **اجرای الگوریتم توسط یک یا چند مجری (انسان، ماشین و یا حتی حیوان).**
- **رسیدن به آن‌چه به عنوان نتیجه موردنظر بوده است.**

**مثال: روش (الگوریتم)هایی برای برطرف کردن مشکلات اقتصادی توسط متخصصین اقتصاد (طراحان الگوریتم).**

**اجرا توسط مردم و مسئولین (مجریان الگوریتم).**

**رفع مشکلات اقتصادی موردنظر (نتیجه)، درصورت درستی روش‌های ارائه شده، و اجرای درست آن.**

**عدم اطلاع مجریان الگوریتم از چگونگی رفع مشکلات.**

**مثال دیگر: عملیات در سیرک‌ها یا محل‌های نمایش حیوانات دریایی مثل دلفین‌ها**

**لزوم رعایت خصوصیات زیر در طراحی الگوریتم برای این‌که قابل‌درک و اجرا باشد:**

- **استفاده از زبان قابل فهم برای‌مجری (انسان،حیوان یا ماشین).**
- **لزوم وجود جزئیات کافی به نحوی که برای مجری به راحتی قابل درک و قابل اجرا باشد.**
- **تعیین دقیق ترتیب انجام عملیات**
- **لزوم وجود شرط خاتمه**

### الگوریتم درمان

**مسئله: بیماری به پزشک مراجعه نموده است و اظهار می‌دارد که احساس می‌کند تب شدیدی دارد و علاوه بر آن، سر او هم خیلی زیاد درد می‌کند.**

**الگوریتم: پزشک پس از معاینه وی، برای او نسخه‌ای می‌نویسد و به او می‌گوید که باید به مدت هفت روز، روزانه سه‌بار و هربار یک قاشق مرباخوری از داروی داده شده را در یک لیوان آب حل نموده و بعد میل نماید.**

**طراح و مجری: طراح این الگوریتم پزشک و مجری آن شخص بیمار است.**

**نتیجه: درصورتی‌که تشخیص درست بوده، الگوریتم نیز صحیح باشد و مجری آن را درست فهمیده باشد و به ترتیب گفته شده اجرا نماید آن‌گاه بیماری رفع خواهد گردید.**

**موارد چهارگانه بالا در این مثال: زبان قابل فهم، جزئیات کافی، ترتیب انجام عملیات، شرط خاتمه.**

## توانایی‌های کامپیوتر

**اجرای همه الگوریتم‌های مورد بحث ما توسط کامپیوتر، توانایی‌های اصلی کامپیوتر به عنوان یک مجری:**

- **خواندن داده‌های مورد نیاز از محیط خارج و ذخیره آن‌ها در حافظه.**
- **انجام عملیات محاسباتی روی داده‌های ذخیره شده در حافظه و برگرداندن نتایج حاصل به حافظه.**
- **انجام عملیات منطقی (مثل مقایسه) روی مقادیر موجود در حافظه و ثبت کردن نتیجه آنها در حافظه.**
- **اعلام مقادیر ذخیره شده در حافظه به صورت‌های مختلف (طبق درخواست) به محیط خارج.**
- **انجام عملیات با هر درجه‌ای از پیچیدگی و با هر تعداد دفعات تکراری که لازم باشد با دقت و سرعت مشخص (که معمولاً بسیار بالا می‌باشد) بدون این‌که درجه پیچیدگی یا دفعات تکرار تأثیری روی نتیجه کار داشته باشد.**

**توجه: ماشین هیچ مرحله‌ای را، هرچند هم جزئی، به صورت ذاتی انجام نمی‌دهد و باید به آن گفته شود.**

## الگوریتم جمع

**مراحل دقیق جمع زدن تعدادی عدد با ماشین حساب:**

*اول ماشین حساب پاک می‌شود. بعد هر عدد همراه با علامت جمع روی آن زده می‌شود که در نتیجه عدد وارد شده وسپس حاصل‌جمع نمایش داده می‌شود. پس از خاتمه اعداد نتیجه نهایی یعنی حاصل‌جمع همه عددها روی پنجره ماشین حساب قابل دیدن است.*

## الگوریتم جمع تعدادی عدد با استفاده از شرح فوق

**۱- حاصل جمع را صفر کن.**

**۲- یک عدد از ورودی دریافت کن.**

**۳- اگر عدد صفر است به مرحله ۷ برو.**

**۴- عدد را به حاصل‌جمع اضافه کن.**

**۵- عدد را در خروجی قرار بده.**

**۶- از مرحله ۲ تکرار کن.**

**۷- حاصل‌جمع را در خروجی قرار بده.**

**۸- خاتمه کار.**

**موارد مهم: وجود حلقه تکرار**

**چگونگی بررسی کردن خاتمه اعداد**

**نام‌های عدد و حاصل‌جمع برای مقادیر**

## نکات مهم در نوشتن یک الگوریتم

**۱- مراحل الگوریتم باید با ترتیب درست و منطقی نوشته شود.**

**۲- برای مقادیر باید نام‌های با معنی انتخاب کرد. (خانه‌های حافظه، متغیر یا شناسه).**

**۳- کلیه حلقه‌های تکرار باید پایان‌پذیر باشند (وجود شرایط اجرا، توقف تکرار)**

**۴- الگوریتم نیز به نوبه خود باید پایان پذیر باشد.**

**۵- جملات این‌گونه الگوریتم‌ها به زبان‌های برنامه‌نویسی سطح بالا نزدیک باشد.**

**به این ترتیب الگوریتم جمع تعدادی عدد به شکل زیر درمی‌آید.**

**۱- دستور** JAM ← ۰ **را اجرا کن.**

**۲- یک مقدار از ورودی دریافت کن و در متغیر** ADAD **قرار بده.**

**۳- اگر** ADAD = ۰ **است آن‌گاه به دستور شماره ۷ برو.**

**۴- دستور** JAM ← JAM + ADAD **را اجرا کن.**

**۵- محتوای** ADAD **را در خروجی قرار بده.**

**۶- به دستور شماره ۲ برو.**

**۷- محتوای** JAM **را در خروجی قرار بده.**

**۸- خاتمه کار.**

## انواع دستورها در الگوریتم

**۱- دستورهای محاسبه و تخصیص: (دستورهای با شماره ۱ و ۴ ).**

**۲- دستورهای ورودی: (دستور شماره ۲ ).**

**۳- دستورهای خروجی: (دستورهای شماره ۵ و ۷ ).**

**۴- دستورهای کنترل اجرا: (دستورهای شماره ۶ و ۸ )**

**۵- دستورهای شرطی: (دستور شماره ۳)**

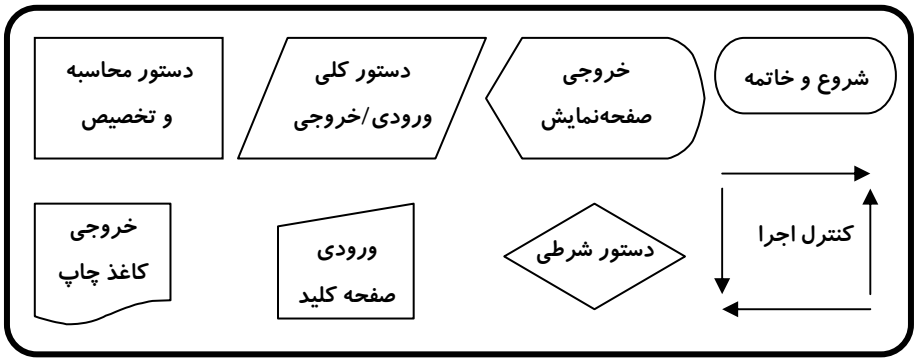
**تشکیل حلقه تکرار با ترکیبی از دستورها (در این مثال ۲، ۳، ۴، ۵ و ۶)**

## ۲-۲-۳ رسم روندنما

**اشکالات زبان‌های محاوره‌ای: وجود ابهام، قابل فهم برای افرادی که زبان را می دانند.**

**روندنما: بیان الگوریتم با استفاده از یک زبان تصویری.**

**نمایش انواع دستورها با شکل‌های مختلف هندسی.**



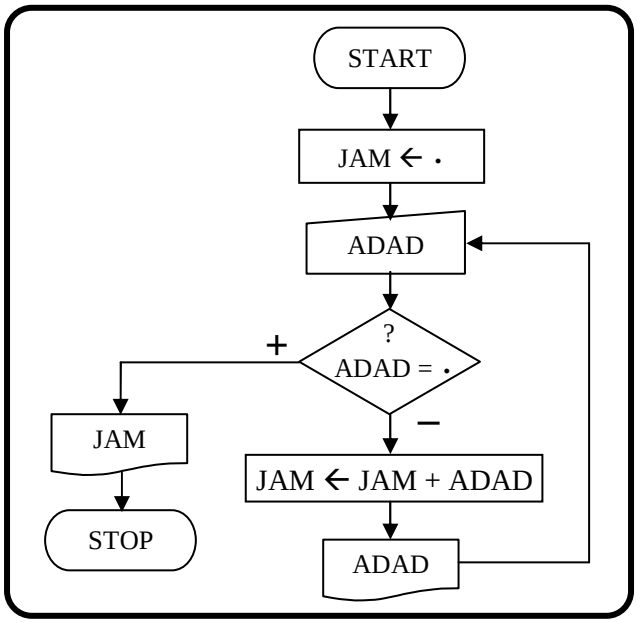

**شکل ۲-۱: انواع شکل‌های مهم مورد استفاده در رسم روندنما.**

### روندنمای جمع تعدادی عدد

**عمومیت بخشیدن به الگوریتم یا روندنما**

**روندنما برای حل فقط یک معادله درجه دوم (شکل ۲-۷)**

**حالت کلی‌تر: روندنما برای حل تعدادی معادله درجه اول یا درجه دوم (شکل ۲-۸).**




**شکل ۲-۲: روندنمای جمع تعدادی عدد.**

## بهینه‌سازی الگوریتم‌ها

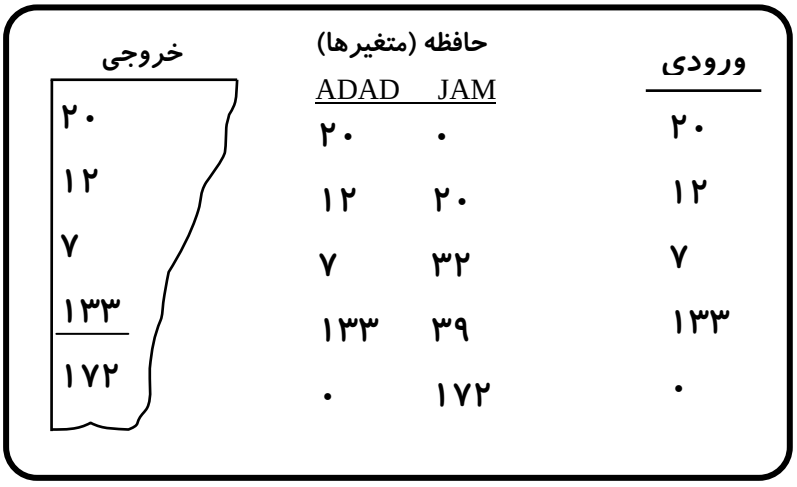
**معیارهای بهتربودن یک الگوریتم از بین چند الگوریتم:**

- **سادگی و قابل فهم بودن الگوریتم برای افرادی غیر از طراح آن.**
- **سهولت تبدیل الگوریتم به یک حالت کلی.**
- **کم‌بودن نسبی حجم حافظه مورد نیاز و حجم عملیات آن.**
- **کم‌بودن نسبی تعداد اسامی متغیرهای استفاده شده در آن برای ذخیره مقادیر در آنها.**
- **در خیلی از موارد کوتاه کردن الگوریتم باعث پیچیده شدن منطق آن خواهد**
- **وجود یک منطق ساده و قابل فهم اولین اصل بهینه‌بودن یک الگوریتم می‌باشد.**

## انواع حلقه‌های تکرار

- **حلقه‌های تکرار با شرط خاص برای خاتمه (شکل ۲-۸، حل تعدادی معادله درجه دوم).**
- **حلقه‌های تکرار با شمارنده، شمارنده، مقدار اولیه، قدرنسبت و نهایی (جمع زدن n عدد، شکل ۲-۶)**
- **هم شمارنده و هم شرط خاص (شکل ۲-۱۲ محاسبه سری تا یک تعداد جملات یا یک تقریب).**

## ۲-۲-۴ بررسی درستی روندنما با استفاده از داده‌های آزمایشی




**شکل ۲-۳: نتیجه دنبال‌کردن روندنمای جمع تعدادی عدد.**

## ۲-۲-۵ نوشتن برنامه با استفاده از یک زبان برنامه‌نویسی

**مجری الگوریتم و روندنما انسان می‌باشد و هیچ‌کدام برای کامپیوتر قابل فهم نیستند.**

**باید آن را به یکی از زبان‌های قابل فهم کامپیوتر(زبان‌های برنامه‌نویسی) تبدیل نمود.**

**انجام برنامه‌نویسی کامپیوتر یا برنامه‌سازی کامپیوتر.**

**برنامه به زبان ماشین.**

**برنامه به زبان‌های سطح بالا، نیاز به ترجمه به زبان ماشین توسط برنامه مترجم (کامپایلر) و سپس اجرا.**

**شکل ۲-۴: برنامه معادل الگوریتم جمع تعدادی عدد به زبان C.**

**تفاوت این برنامه با برنامه بخش ۱-۴-۴ از فصل اول (جدول ۱-۳) به زبان ماشین.**

**شباهت دستورهای این برنامه به جمله‌های زبان محاوره‌ای.**

**توجه به تشابه ساختاری بین برنامه و روندنما.**

**ارتباط بین دستورهای برنامه و جعبه‌های روند نما.**

```
#include <stdio.h>

main()                                  /* برنامه جمع تعدادی عدد */
{                                       /* شروع دستورهای برنامه */
   int jam, adad;                       /* تعریف اسامی مورد استفاده در برنامه */

   jam = 0;                             /* صفرکردن حاصل‌جمع */
   scanf("%d", &adad);                  /* خواندن اولین عدد از ورودی */
   while (adad != 0)                    /* حلقه تکرار خواندن سایر عددها و بررسی پایان مقادیر ورودی */
   {  jam = jam + adad;                 /* افزودن عدد خوانده شده به حاصل‌جمع */
      printf("%d\n", adad);             /* چاپ عدد خوانده شده */
      scanf("%d", &adad);               /* خواندن عدد بعدی از ورودی */
   }                                    /* خاتمه دستورهای حلقه تکرار */
   printf("%d\n", jam);                 /* چاپ حاصل‌جمع */
   return (0);                          /* توقف برنامه */
}                                       /* خاتمه دستورهای برنامه */
```

**شکل۲-۴: متن برنامه جمع تعدادی عدد، مرحله پنجم حل یک مسئله با استفاده از کامپیوتر.**

## ۲-۲-۶ واردکردن برنامه به کامپیوتر، آزمایش برنامه و استفاده از برنامه

**نیاز به نرم‌افزار کامپایلربرای زبان سطح بالایی که برنامه به آن زبان نوشته شده است (در مورد ما زبان C)**

**امکان وجود اشتباهات دستوری و لزوم رفع آنها.**

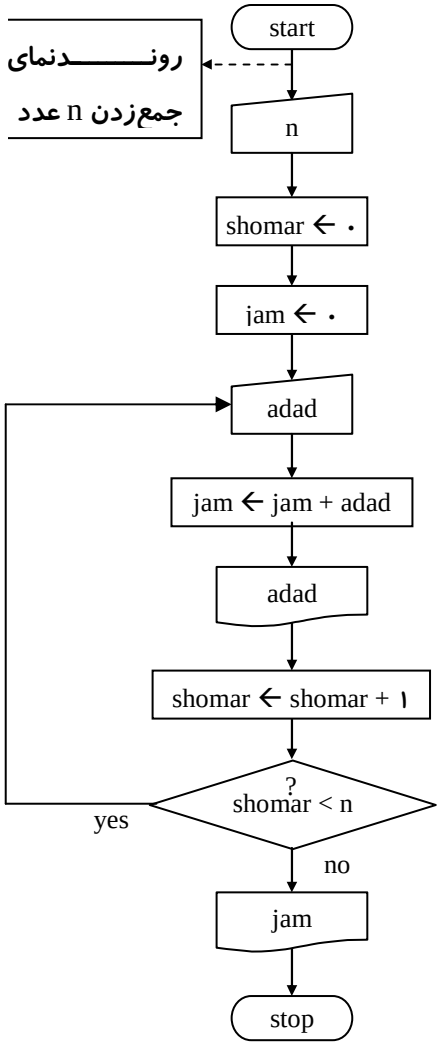
**لزوم دادن داده‌های آزمایشی به عنوان ورودی و بررسی نتیجه.**

```
#include <stdio.h>

main()                                  /* برنامه جمع تعدادی عدد */
{                                       /* شروع دستورهای برنامه */
   int jam, adad;                       /* تعریف اسامی مورد استفاده در برنامه */

   jam = 0;                             /* صفرکردن حاصل‌جمع */
   scanf("%d", &adad);                  /* خواندن اولین عدد از ورودی */
   while (adad != 0)                    /* حلقه تکرار خواندن سایر عددها و بررسی پایان مقادیر ورودی */
   {  jam = jam + adad;                 /* افزودن عدد خوانده شده به حاصل‌جمع */
      printf("%d\n", adad);             /* چاپ عدد خوانده شده */
      scanf("%d", &adad);               /* خواندن عدد بعدی از ورودی */
   }                                    /* خاتمه دستورهای حلقه تکرار */
   printf("%d\n", jam);                 /* چاپ حاصل‌جمع */
   return (0);                          /* توقف برنامه */
}                                       /* خاتمه دستورهای برنامه */

داده‌های ورودی:        ۰  ۱۳۳  ۷  ۱۲  ۲۰

نتایج خروجی برنامه:    ۲۰
12
7
133

172
```

**شکل ۲-۵: نتیجه اجرای برنامه جمع تعدادی عدد، مراحل ششم و هفتم حل یک مسئله با استفاده از کامپیوتر.**

## ۲-۳ روندنماهای نمونه

**روندنمای ۲-۱: الگوریتمی بنویسید که نخست مقدار n را از ورودی خوانده و سپس به تعداد n عدد از ورودی بخواند و پس از چاپ آن عددها در خروجی، حاصل‌جمع آن‌ها را نیز چاپ نماید. الگوریتم خود را برای یک مجموعه داده‌های آزمایشی مناسب دنبال کرده و سپس آن‌را به یک روندنما تبدیل نمایید.**




**الگوریتم جمع‌زدن n عدد**

۱۰- n را از ورودی دریافت کن.
۲۰- دستور shomar ← ۰ را اجرا کن.
۳۰- دستور jam ← ۰ را اجرا کن.
۴۰- adad را از ورودی دریافت کن.
۵۰- دستور jam ← jam + adad را اجرا کن.
۶۰- محتوای adad را در خروجی قرار بده.
۷۰- دستور shomar ← shomar + ۱ را اجرا کن.
۸۰- اگر shomar < n به دستور شماره ۴۰ برو.
۹۰- محتوای jam را در خروجی قرار بده.
۱۰۰- خاتمه کار

**نتیجه دنبال‌کردن الگوریتم**

**حافظه (متغیرها)**

| n | shomar | jam | adad | ورودی |
|---|---|---|---|---|
| ۵ | ۰ | ۰ | ۲۰ | ۵ |
| | ۱ | ۲۰ | ۱۲ | ۲۰ |
| | ۲ | ۳۲ | -۷ | ۱۲ |
| | ۳ | ۲۸ | | -۷ |

**خروجی**

۲۰
۱۲
-۷

**شکل ۲-۶: الگوریتم و روندنمای ۲-۱، جمع‌زدن n عدد همراه با نتیجه دنبال‌کردن آن.**

**روندنمای ۲-۲: روندنمایی رسم کنید که در آن نخست ضرایب یک معادله درجه دوم از ورودی خوانده شود و پس از چاپ ضرایب، معادله حل شده و با توجه به علامت $\Delta$ نتیجه در قالب ریشه‌های مختلف، ریشه مضاعف و بدون ریشه حقیقی همراه با توضیحات مناسب چاپ گردد.**

**الگوریتم حل معادله درجه دوم**

۱- ضرایب معادله را از ورودی بخوان و به ترتیب در خانه‌های A، B و C قرار بده.

۲- محتویات A، B و C را در خروجی چاپ کن.

۳- دستور $D \leftarrow B^2 - 4 \times A \times C$ را اجرا کن.

۴- اگر $D < 0$ است پیغام "No Real Root!" را چاپ کرده و سپس توقف کن.

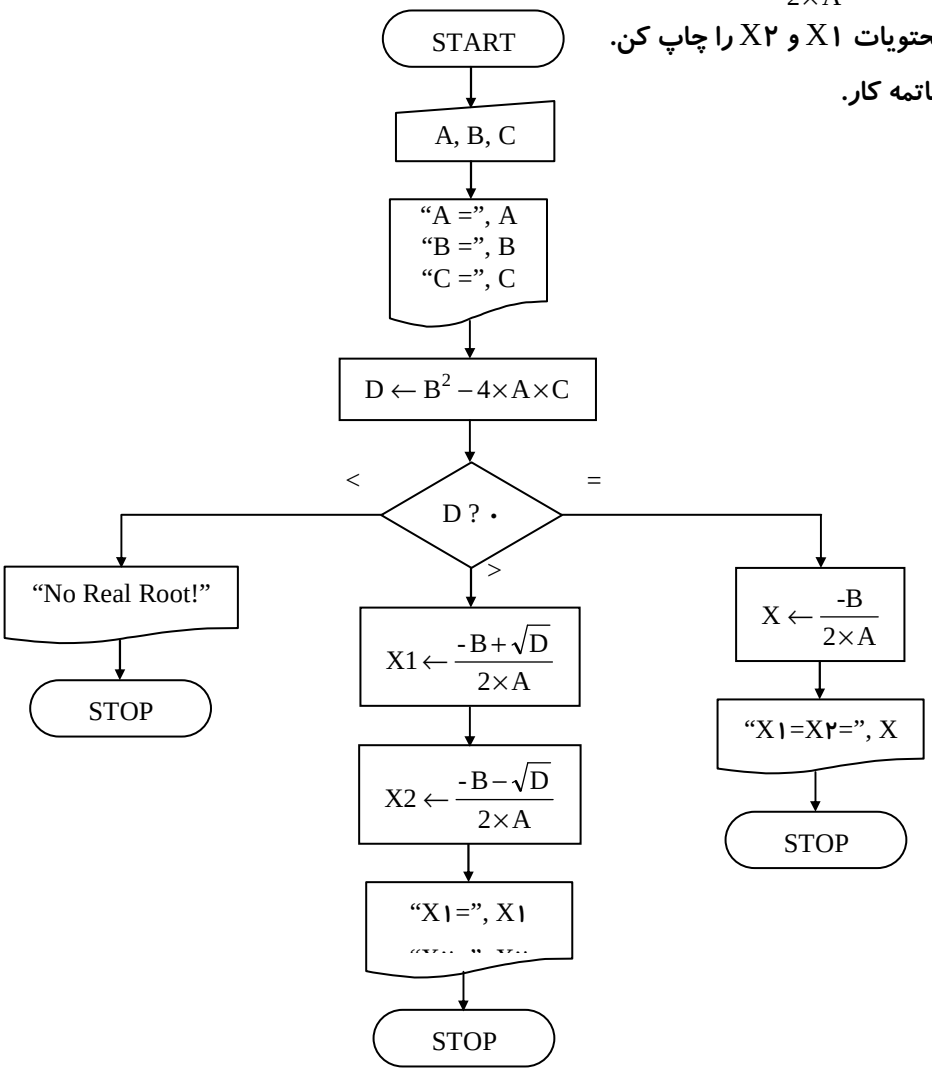
۵- اگر $D = 0$ است دستور $X \leftarrow \frac{-B}{2 \times A}$ را اجراکن، محتوای X را چاپ کن و سپس توقف کن.

۶- دستور $X1 \leftarrow \frac{-B + \sqrt{D}}{2 \times A}$ را اجراکن.

۷- دستور $X2 \leftarrow \frac{-B - \sqrt{D}}{2 \times A}$ را اجراکن.

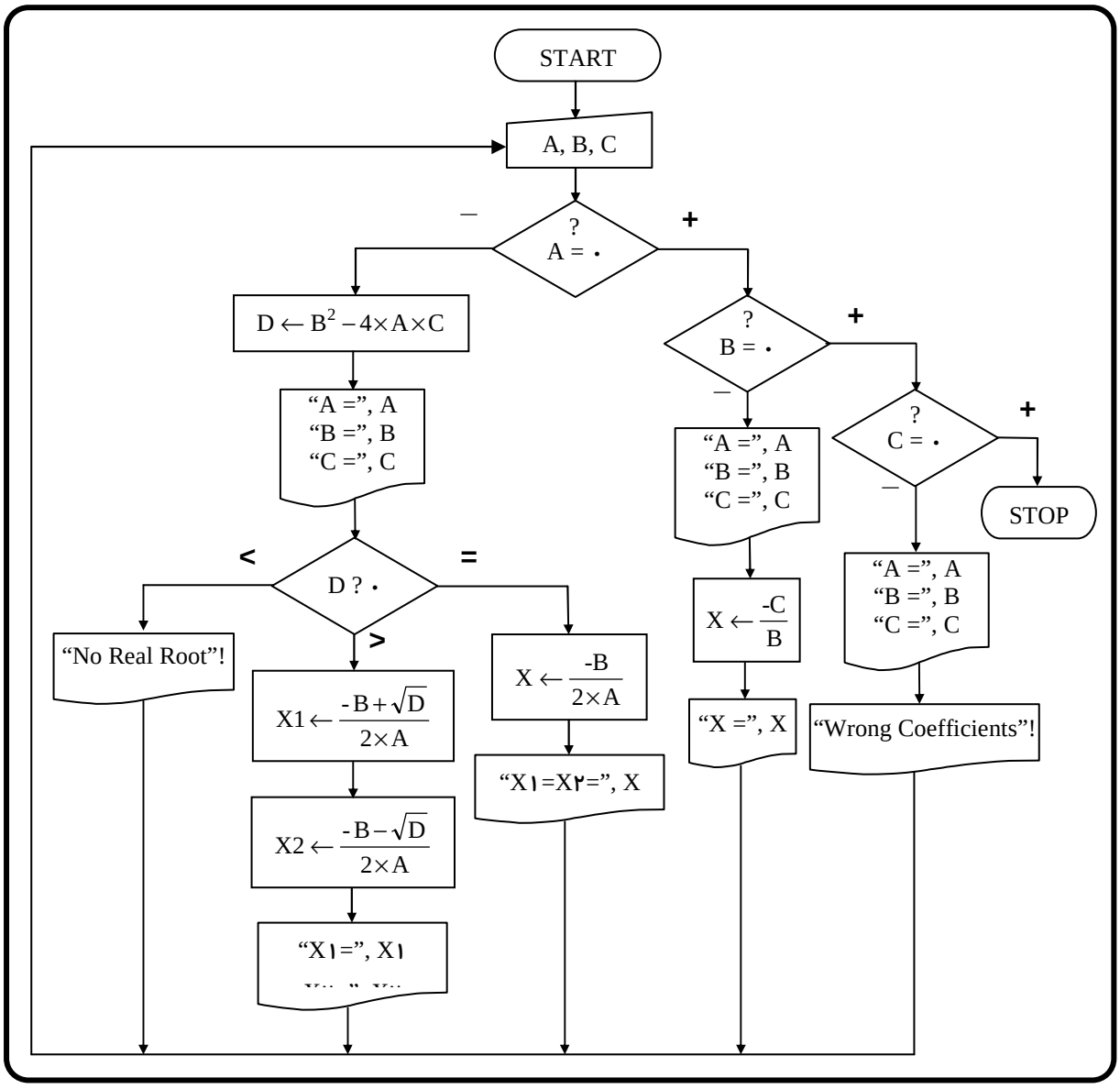
۸- محتویات X۱ و X۲ را چاپ کن.

۹- خاتمه کار.



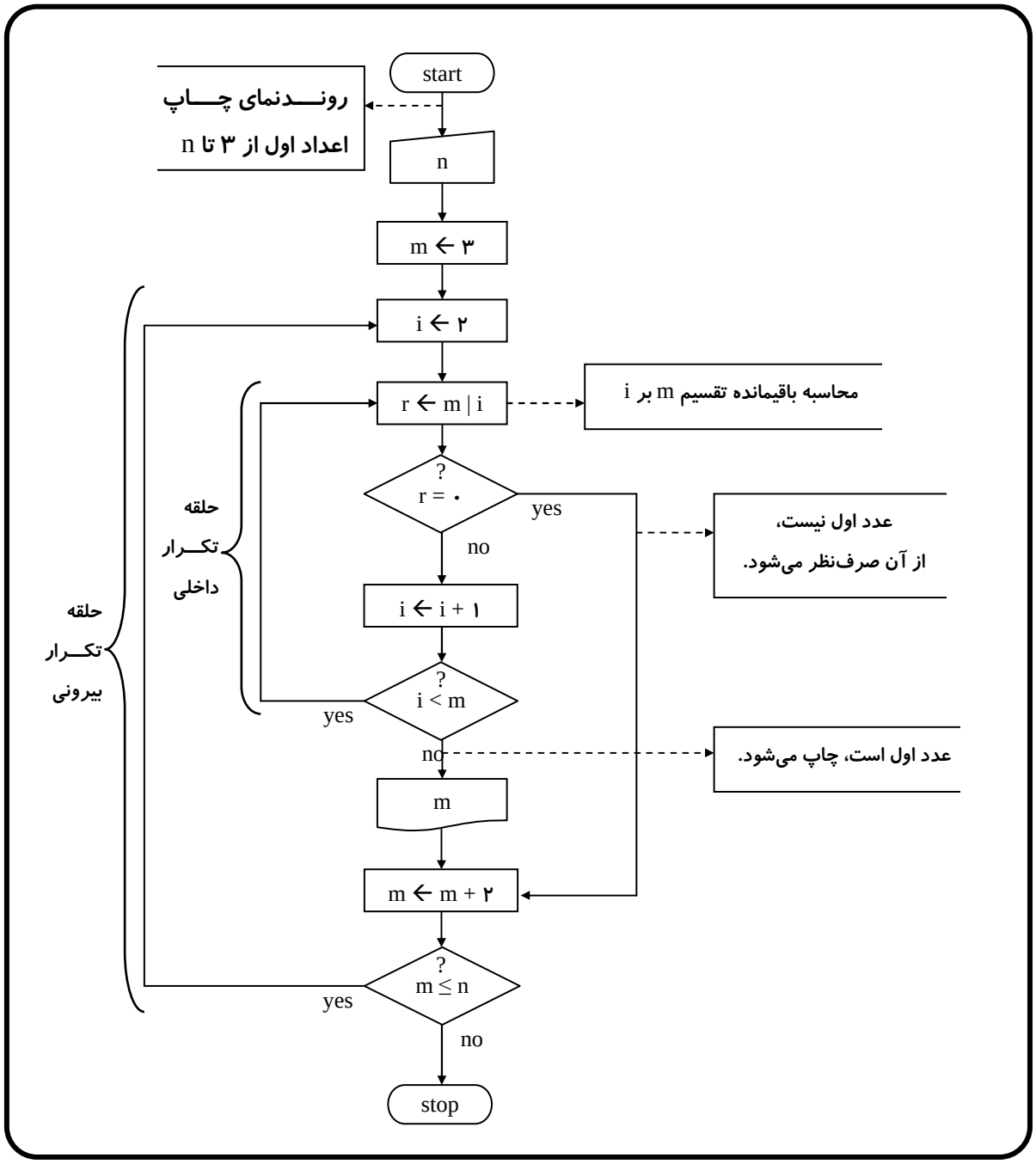

**شکل ۲-۷: الگوریتم و روندنمای ۲-۲، حل یک معادله درجه دوم.**

**روندنمای ۲-۳: مسئله حل یک معادله درجه دوم که در روندنمای ۲-۲ رسم شده را طوری اصلاح کنید که اولاً ضرایب تعدادی معادله را از ورودی بخواند و هر معادله را جداگانه حل کند با ایـن شـرط کـه در پایـان داده‌های ورودی به جای ضرایب سه مقدار صفر داده شده باشد. ثانیاً درصورتی که مقدار A صفر بود یعنـی معادله درجه اول بود، جواب آن معادله درجه اول را نیز محاسبه و چاپ نماید و اگر ضرایب غلط بود یعنیA و B صفر بودند ولی C غیرصفر بود، پیغام مناسب چاپ کند و از آن ضرایب صرف‌نظر نماید.**



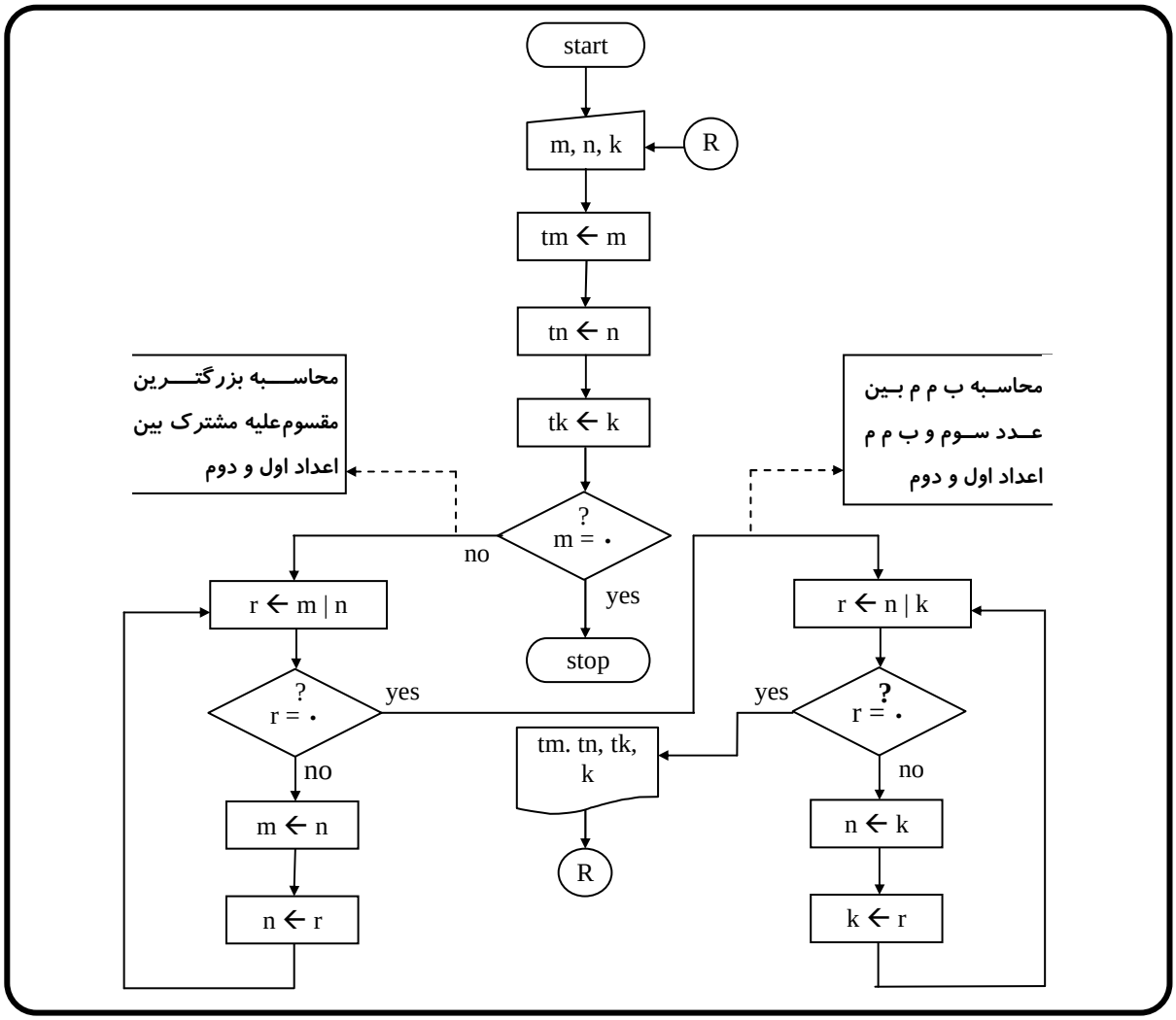

**شکل ۲-۸: روندنمای ۲-۳، حل تعدادی معادله درجه اول یا دوم.**

**روندنمای ۲-۴: روندنمایی رسم کنید که در آغاز عدد صحیح و مثبت n را از ورودی بخوانـد و سـپس کلیـه اعداد اول بین ۳ و n را محاسبه کرده و چاپ نماید.**



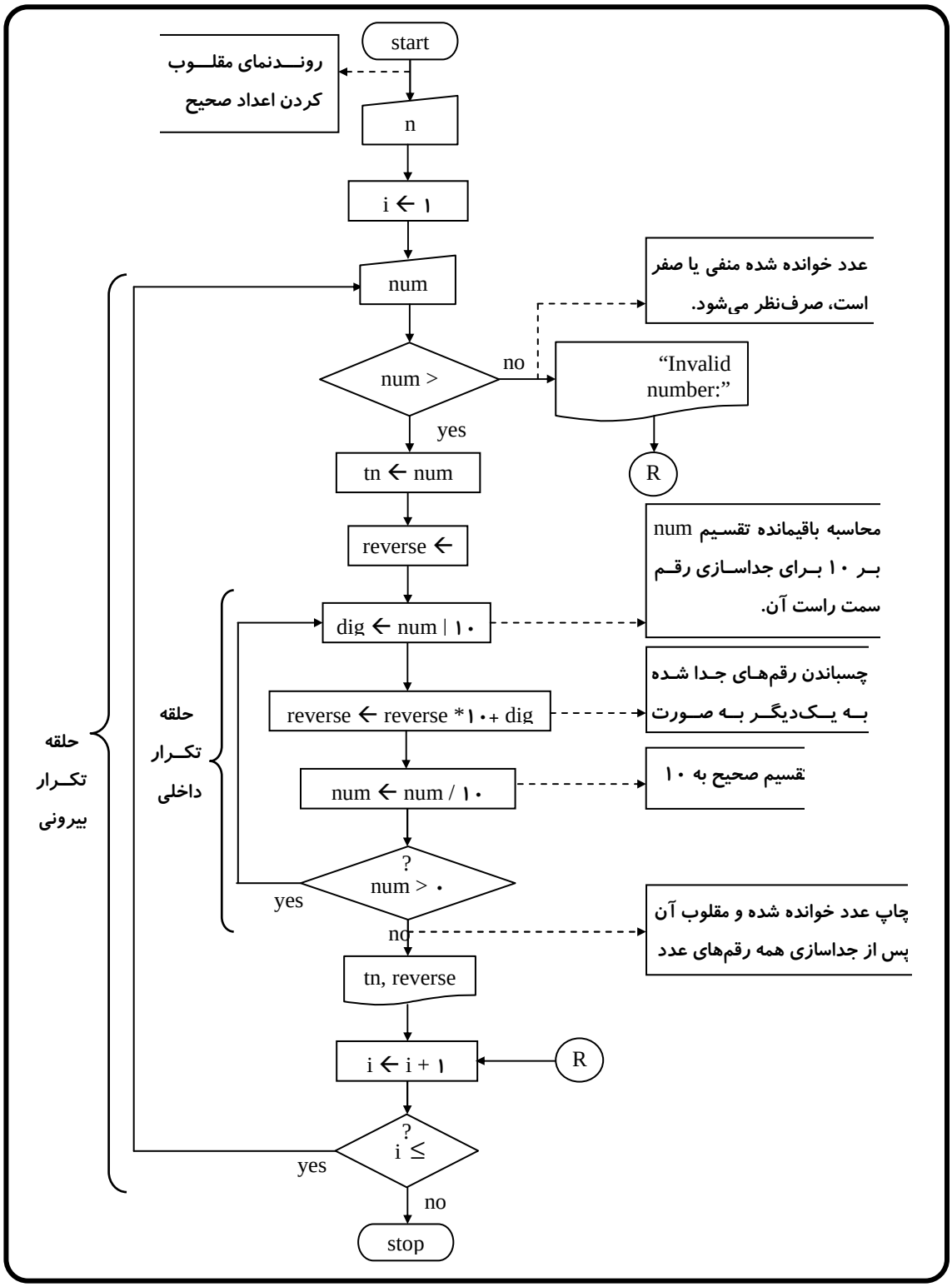

**شکل ۲-۹: روندنمای ۲-۴، چاپ اعداد اول از ۳ تا n.**

**روندنمای ۲-۵: روندنمایی رسم کنید که در یک حلقه تکرار هر بار سه عدد صحیح و مثبت را از ورودی بخواند و پس از چاپ آن سه عدد، بزرگترین مقسوم‌علیه مشترک آن‌ها را محاسبه کرده چاپ نماید. توجه داشته باشید که اعداد در ورودی هیچ‌گونه ترتیبی از نظر بزرگی و کوچکی ندارند و پایان داده‌های ورودی نیز با سه عدد صفر مشخص شده است.**



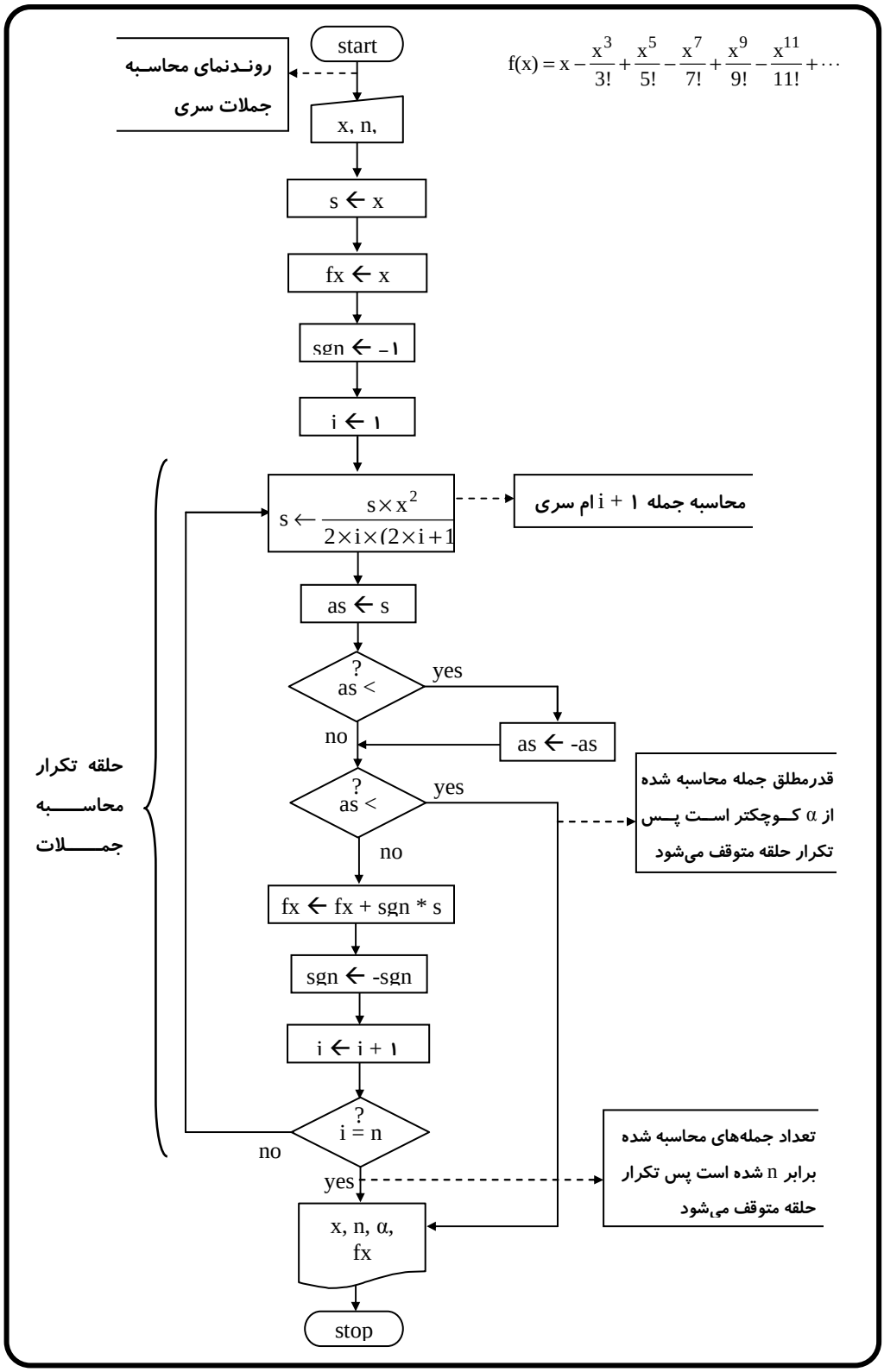

**شکل ۲-۱۰: روندنمای ۲-۵، محاسبه بزرگترین مقسوم‌علیه مشترک سه عدد.**

**روندنمای ۲-۶: تعدادی عدد صحیح در دست است، روند نمایی رسم کنید که هر عدد را خوانده درصورتی‌که مثبت باشد مقلوب عدد را ساخته و همراه با خود عدد در خروجی چاپ نماید، در غیراین‌صورت عدد خوانده شده را همراه با پیغام مناسب چاپ کرده و از آن صرف‌نظر نماید. تعداد عددهای مزبور در شروع داده‌ها قرار دارد. لازم به توضیح است که مقلوب یک عدد، عددی است که ترتیب رقم‌هایش برعکس ترتیب رقم‌های عدد اولیه باشد.**



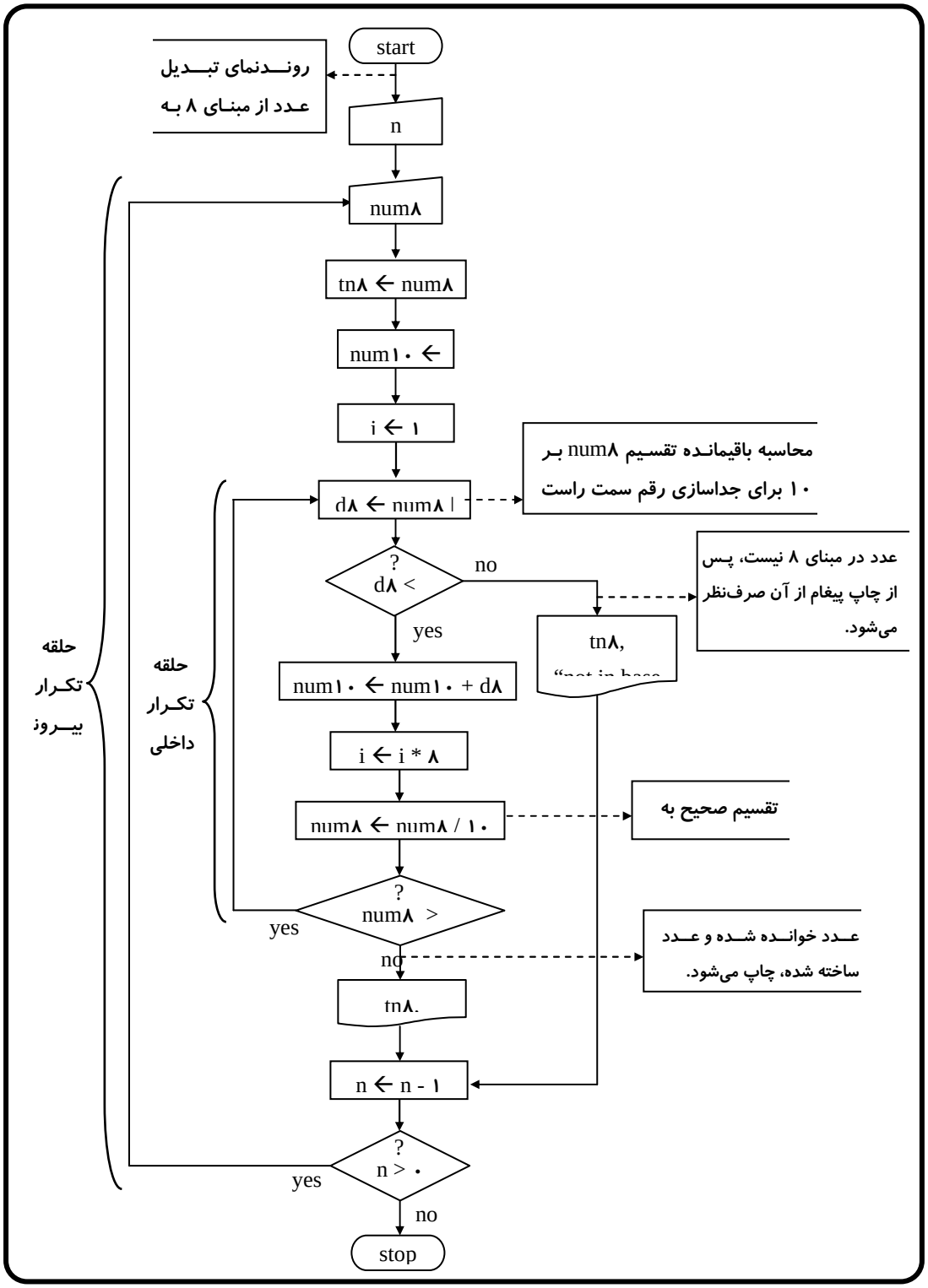

**شکل ۲-۱۱: روندنمای ۲-۶، مقلوب کردن اعداد صحیح.**

**روندنمای ۲-۷: فرمول یک سری در پایین داده شده است، روندنمایی رسم کنید که مقادیر $x$، $n$ و $\alpha$ را از ورودی بخواند و مقدار $f(x)$ را محاسبه نماید. محاسبه تا جایی ادامه یابد که یا تعداد جمله‌های محاسبه شده برابر $n$ شود یا این‌که قدرمطلق جمله محاسبه شده از $\alpha$ کم‌تر باشد. در پایان مقادیر $x$، $n$ و $\alpha$ را همراه با مقدار $f(x)$ در خروجی چاپ نماید.**



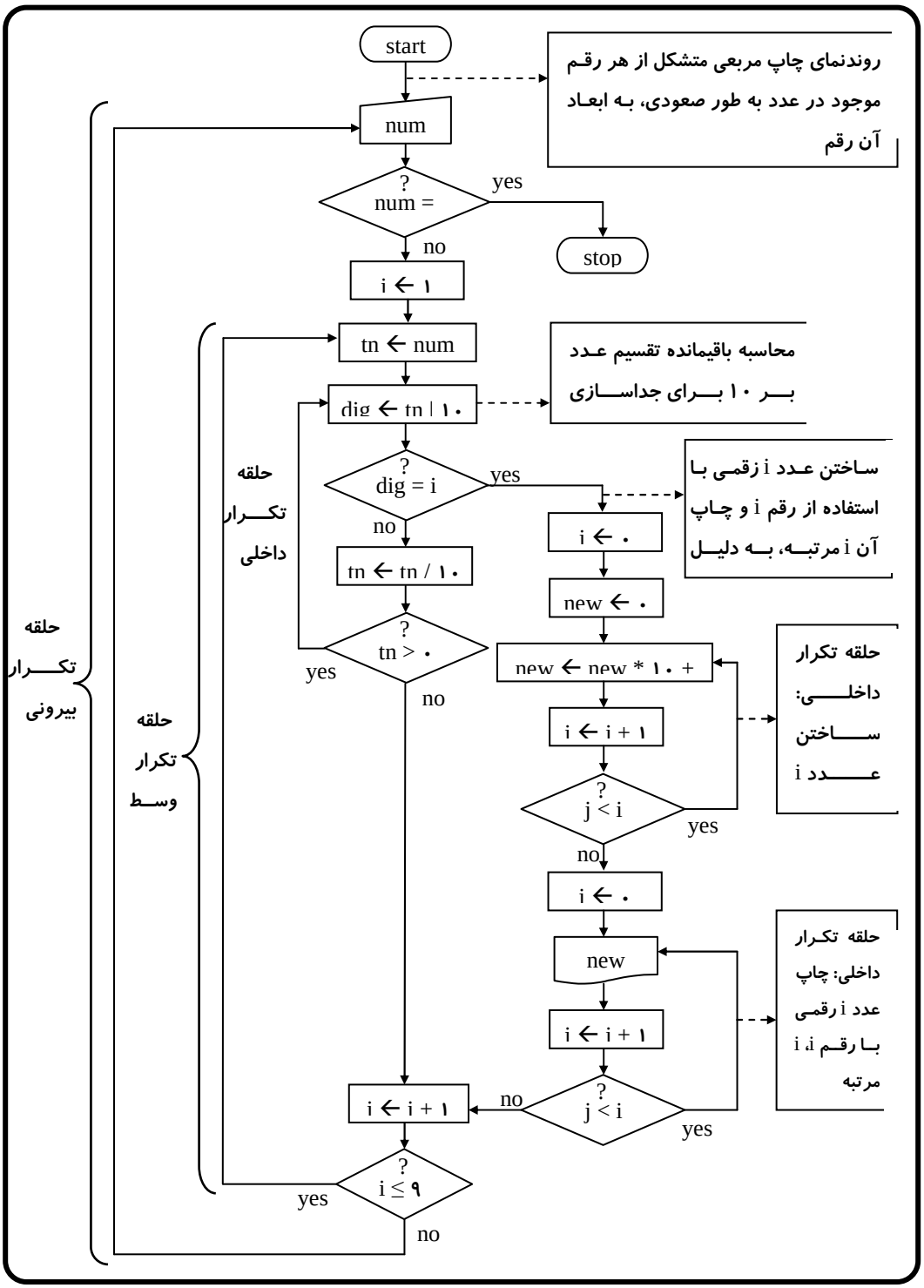

**شکل ۲-۱۲: روندنمای ۲-۷، محاسبه جملات سری.**

**روندنمای ۲-۸: روندنمایی رسم کنید که تعدادی عدد صحیح و مثبت را از ورودی بخواند و برای هر مقدار خوانده شده، درصورتی‌که آن مقدار می‌تواند یک عدد در مبنای هشت باشد آن‌را به مبنای ده برده و همراه با عدد اولیه چاپ نماید. در غیراین‌صورت عدد را همراه با پیغام مناسب چاپ کند. فرض کنید تعداد عددهایی که باید خوانده شود قبل از آن عددها به عنوان داده ورودی ارائه شود.**




**شکل ۲-۱۳: روندنمای ۲-۸، تبدیل عدد از مبنای ۸ به ۱۰.**

**روندنمای ۲-۹: روندنمایی رسم نمایید که متناوباً عـددی صـحیح و مثبـت را ازورودی بخوانـد ( پایـان عـددهـا بـا عـدد صـفر مشخص شده است) و سپس برای رقم‌های یک تا نه به ترتیب صعودی، درصورتی‌که آن رقـم در بـین رقـم‌هـای عـدد وجـود دارد، عددی بسازد که تعداد رقم‌هایش مساوی رقم مزبور باشد و همه رقم‌های تشکیل دهنده‌اش هم همان رقم باشد. سـپس عدد ساخته شده را به تعداد دفعات آن رقم چاپ نماید. مثلاً اگر عدد خوانده شده ۲۱۷۲۴۵ باشد روندنما بایـد طـوری عمـل کند که نخست عدد یک را یک بار چاپ کند، بعد عدد ۲۲ را دوبار چاپ نماید، سپس عدد ۴۴۴۴ را چهار مرتبه و . . ..**




**شکل ۲-۴: روندنمای ۲-۹، چاپ مربعی متشکل از هر رقم موجود در عدد به طور صعودی، به ابعاد آن**

## ۲-۴ تکلیف شماره دو

**پرسش *۲-۱۳- تعدادی عدد صحیح مثبت در دست است و تعداد آنها قبل از همه عـددها مشـخص شـده است، روندنمایی رسم کنید که هر عدد را خوانده و موارد زیر را انجام دهد.**

**مقلوب عدد خوانده شده، مجموع فاکتوریل رقم های فرد عدد و مجموع فاکتوریل های زوج عدد را جداگانه یافته و همراه با خود عدد چاپ نماید.**

**از روی عدد خوانده شده عدد جدیدی بسازد که ارقام آن همان ارقام عدد اولیه باشد ولی به ترتیـب نزولـی مرتب شده باشد و در مورد رقم های تکراری نیز رقم مزبور فقط یک بار در عدد جدید وجود داشته باشـد. مثلاً اگر عدد خوانده شده ۳۷۱۰۶۳۲ باشد عدد جدید ۷۶۳۲۱۰ خواهد بود.**

**نهایتاً اگر تعداد رقم های عدد اولیه زوج است، از روی آن عدد، عدد دیگری به این صورت بسازد کـه رقـم اول و دوم، رقم سوم و چهارم و به همین ترتیب تا رقم ماقبل آخر و رقم آخر جایشان با هـم تعـویض شـده باشد و عدد اولیه را همراه با عدد شناخته شده چاپ نماید. مثلاً اگر عدد خوانـده شـده از ورودی ۱۴۳۷۹۱ باشد عدد جدید ۴۲۷۳۱۹ خواهد بود.**