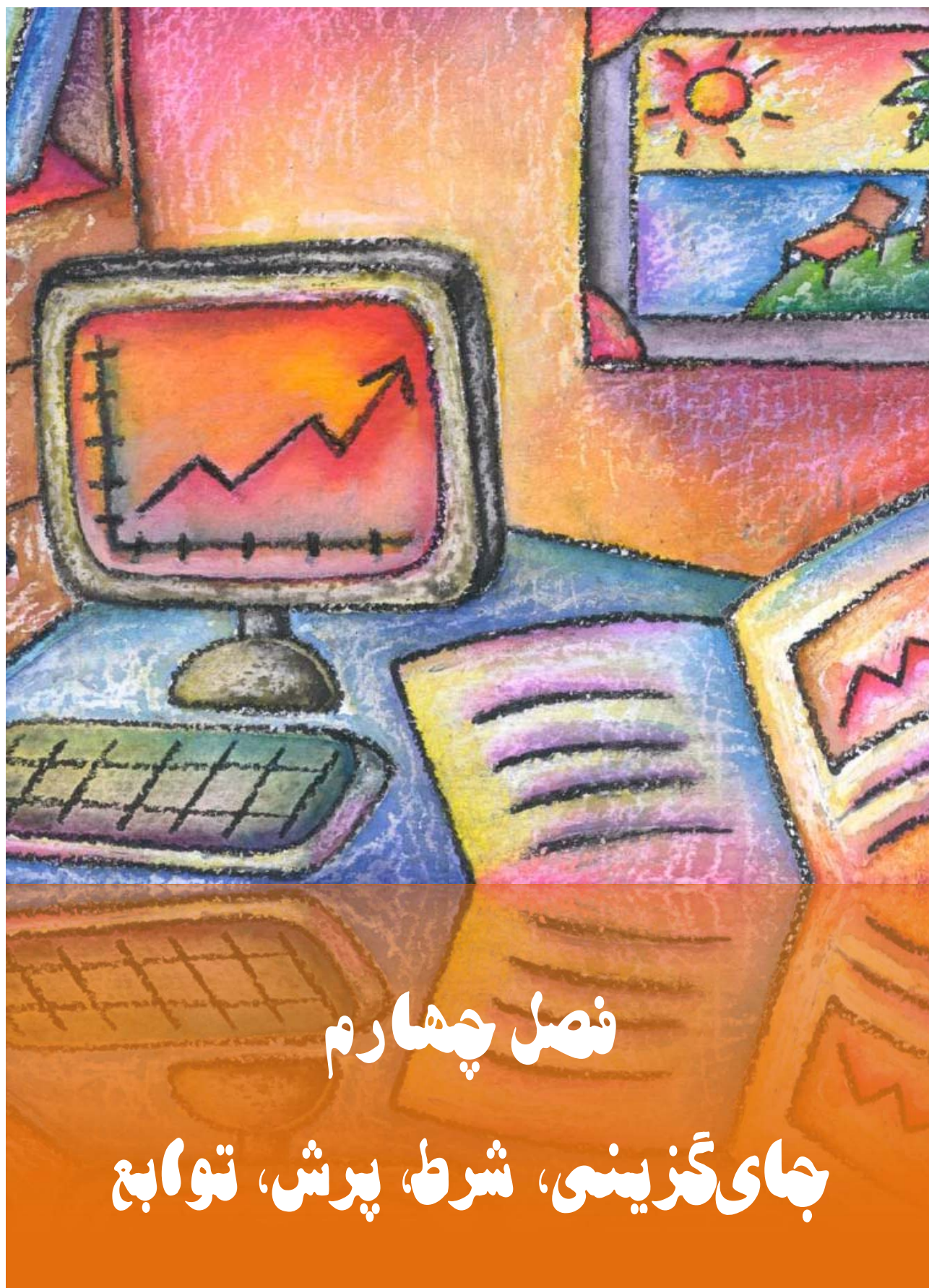




۴.۱. مقدمه

در این فصل با چند دستور پرکاربرد دیگر در زبان بیسیک آشنا می‌شویم. دستور جایگزینی برای قرار دادن یک مقدار درون یک خانه حافظه (بدون پرسیدن از کاربر) مورد استفاده قرار می‌گیرد. هم چنین دستورات شرطی را برای گذاشتن شرط در برنامه‌هایمان یاد می‌گیریم. در آخر فصل هم با تعدادی از توابع پرکاربرد در زبان بیسیک آشنا می‌شویم.

.....

۴.۲. دستور جایگزینی (LET)

قبلاً در دستور INPUT با گرفتن یک مقدار از کاربر و قرار دادن آن درون یک خانه حافظه آشنا شده‌اید. اکنون این سؤال پیش می‌آید که اگر بخواهیم یک مقدار معلوم (که آن را از کاربر نمی‌پرسیم) درون یک خانه حافظه بریزیم، چه باید بکنیم.

برای این منظور از دستور LET استفاده می‌کنیم. شکل کلی این دستور به صورت زیر است:

عبارت = نام خانه حافظه LET

این دستور حاصل عبارت سمت راست را حساب کرده و آن را درون خانه حافظه (متغیر) مشخص شده می‌ریزد.

 **توجه:** در دستور LET همواره نام خانه حافظه در سمت چپ و عبارت در سمت راست نوشته می‌شود. یعنی اگر بخواهیم عدد 2 را درون خانه حافظه A بریزیم، باید بنویسیم: LET A = 2، نه این که بنویسیم: LET 2 = A.

 **نمونه:** چند کاربرد دستور LET :

$$\text{LET } B = 34$$

$$\text{LET } \text{Salam} = 12 * 5 - 6 * 3$$

$$\text{LET } \text{Ali} = (2 + 5) * 3 / 7$$

$$\text{LET } A = B * 2 + 5$$

بعد از اجرای دستورات فوق، محتوای خانه‌های حافظه B، Salam و Ali به ترتیب برابر 34، 42 و 3 می‌شود. همچنین در خانه حافظه A، ۲ برابر مقدار خانه حافظه B به علاوه‌ی ۵ قرار می‌گیرد.

 **توجه:**

۱. دستور $\text{LET } A+B = C+D$ در زبان بیسیک بی معنی است، چون در سمت چپ مساوی حتماً باید اسم یک خانه حافظه بیاید، اما $A + B$ اسم خانه حافظه نیست.

۲. در زبان بیسیک $\text{LET } A = B$ و $\text{LET } B = A$ با هم تفاوت دارند، دستور اول محتوای خانه حافظه B را درون خانه حافظه A می‌ریزد و دستور دوم محتوای خانه حافظه A را درون B می‌ریزد.

۳. وقتی دستور $\text{LET } A = B$ اجرا می‌شود، محتوای خانه حافظه B درون خانه حافظه A ریخته می‌شود، اما محتوای خانه حافظه B خودش دست نخورده باقی می‌ماند. یعنی مثلاً اگر قبل از اجرای این دستور، درون خانه حافظه A عدد 3 و درون خانه حافظه B عدد 5 قرار گرفته باشد، بعد از اجرای دستور، درون هر دو خانه حافظه، عدد 5 قرار دارد.

۴. می‌توانید دستور LET را در زبان بیسیک ننویسید، یعنی دو خط زیر در زبان بیسیک یک معنی دارند:

$$\text{LET } A = 2 * 3$$

$$A = 2 * 3$$

۵. در هر خانه حافظه تنها یک عدد می‌توانیم ذخیره کنیم. یعنی وقتی مقداری را با دستور LET درون یک خانه حافظه می‌ریزیم، مقدار قبلی آن از بین می‌رود.

 **نمونه:** برنامه‌ای بنویسید که محتوای خانه حافظه A را از کاربر بپرسد. سپس آن را ۵ برابر کرده، درون خانه حافظه B بریزد و نهایتاً محتوای خانه حافظه را نمایش دهد.

پاسخ:

```
INPUT A
LET B = A * 5
PRINT B
```

 **نمونه:** برنامه‌ای بنویسید که مقدار x را بگیرد و حاصل عبارت $u^2 + 2u + 4$ را با شرط $u = \frac{1+x}{1-x}$ نمایش دهد.

پاسخ:

```
INPUT x
LET u = (1 + x) / (1 - x)
PRINT u ^ 2 + 2 * u + 4
```

 **نمونه:** برنامه‌ای بنویسید که دو عدد گرفته و در خانه‌های حافظه A و B قرار دهد. سپس محتوای این دو خانه حافظه را با هم عوض کند.

پاسخ: شاید اولین ایده‌ای که به ذهن‌مان برسد به صورت زیر باشد:

```
INPUT A , B
LET A = B
LET B = A
```

اما این برنامه اشتباه است. بیایید خط به خط آن را دنبال کرده و اشکالش را ببینیم. فرض کنید کاربر محتوای خانه‌های حافظه A و B را به ترتیب 2 و 3 وارد کند. پس بعد از اجرای خط اول، محتوای A و

B به صورت زیر است:

A	B
2	3

بعد از اجرای خط دوم، محتوای B درون A ریخته می‌شود، یعنی داریم:

A	B
3	3

بعد از اجرای خط سوم، محتوای A درون B ریخته می‌شود، یعنی داریم:

A	B
3	3

می‌بینید که محتوای خانه حافظه A در خط دوم از بین می‌رود و دیگر به آن دسترسی نداریم تا در B بریزیم. پس برای رفع این مشکل، قبل از خط دوم محتوای A را در جایی دیگر (مثلاً خانه حافظه C) ذخیره می‌کنیم و در خط آخر آن را درون B می‌ریزیم. یعنی برنامه درست به این شکل است:

```
INPUT A , B
LET C = A
LET A = B
LET B = C
```

این برنامه را خط به خط تعقیب کنید و درستی آن را تحقیق کنید.

آیا می‌دانید؟ 

زبان بیسیک برای عوض کردن محتوای دو خانه حافظه با همدیگر یک دستور دارد. این دستور

```
SWAP A , B
```

محتوای دو خانه حافظه A و B را با هم عوض می‌کند.

برای ریختن یک کلمه یا جمله در خانه حافظه از نوع کلمه‌ای، از دستور LET استفاده می‌کنیم و کلمه یا جمله را درون گیومه می‌گذاریم. مثلاً دستور:

```
LET A$ = "SALAM"
```

کلمه‌ی SALAM را درون خانه حافظه A\$ ذخیره می‌کند. باز هم دقت کنید که عبارت داخل گیومه بدون هیچ تغییری در متغیر ذخیره می‌گردد.

.....

۳.۴. دستور شرط

می‌خواهیم برنامه‌ای بنویسیم که با گرفتن طول و عرض یک چهار ضلعی بگوید آن چهار ضلعی مربع است یا مستطیل. مسلماً با دستوراتی که تا به حال خوانده‌ایم این کار امکان ندارد. زبان بیسیک برای مقایسه و شرط هم دستوراتی دارد که در این قسمت آن‌ها را یاد می‌گیریم.

زبان بیسیک دستوری دارد که در صورت درست بودن یک شرط، کاری را انجام می‌دهد و در صورت غلط بودن آن به خط بعدی می‌رود. این دستور به شکل زیر است:

دستور THEN شرط IF

وقتی رایانه به خط بالا می‌رسد، شرط بعد از کلمه IF را بررسی می‌کند. اگر این شرط درست بود، دستور بعد از کلمه‌ی THEN را اجرا می‌کند، و گرنه هیچ کاری انجام نمی‌دهد و به خط بعدی می‌رود.

مثلاً هر گاه رایانه به دستور:

IF A < 2 THEN PRINT "SALAM"

برسد، بررسی می‌کند که آیا محتوای خانه حافظه A کمتر از ۲ است یا خیر. اگر کمتر بود، کلمه SALAM را نمایش می‌دهد و در غیر این صورت کاری نمی‌کند و به خط بعدی می‌رود.

توجه:



شرطی که جلوی دستور IF وجود دارد، غالباً مقایسه بین دو مقدار است، مثلاً:

دستور IF A = B THEN

دستور IF A * 2 < B THEN

دستور IF C * 2 + B > K THEN

علامت‌های مقایسه دیگر در زبان بیسیک به صورت زیر هستند:

علامت در بیسیک				شرط
=<	یا	<=		$\leq$
=>	یا	>=		$\geq$
><	یا	<>		$\neq$



یک دستور جدید :

هر جای برنامه دستور END را بنویسید، اجرای برنامه همان‌جا خاتمه می‌یابد.

حال می‌خواهیم برنامه‌ای که در ابتدای این بخش گفتیم (تشخیص مستطیل یا مربع بودن) را بنویسیم:

```
INPUT Tool , Arz
```

```
IF Tool = Arz THEN PRINT "MORABBAE"
```

```
IF Tool <> Arz THEN PRINT "MOSTATIL"
```

مطلب دیگر این است که بخواهیم در صورت درست بودن شرط، به جای یک دستور، چند دستور

اجرا شود، باید چه کنیم؟

چاره‌ی کار به شکل زیر است:

```
IF شرط THEN
```

```
    دستور ۱
```

```
    دستور ۲
```

```
    ...
```

```
END IF
```

یعنی تمام دستوراتی که می‌خواهیم در صورت درست بودن شرط اجرا شوند را بعد از THEN (هر کدام در یک خط) می‌نویسیم و در آخر کلمه‌ی END IF را می‌نویسیم. رایانه با دستوراتی که بعد از THEN و قبل از END IF نوشته شده‌اند به صورت یک مجموعه رفتار می‌کند، یعنی یا همه آن‌ها را اجرا می‌کند یا هیچ کدام را.

دستورات شرطی زبان بیسیک قابلیت دیگری نیز دارند. فرض کنید می‌خواهید رایانه شرطی را بررسی کند. اگر شرط درست بود دستور شماره ۱ و اگر شرط غلط بود دستور شماره ۲ را اجرا می‌کند. با این که با دو دستور IF می‌توان چنین کاری انجام داد، اما زبان بیسیک دستور ویژه‌ای هم برای این کار دارد که به شکل زیر است:

دستور ۲ ELSE دستور ۱ THEN شرط IF

کلمات IF و THEN و ELSE به همین ترتیب نوشته می‌شوند. اگر شرط جلوی IF درست باشد، دستور ۱ و اگر غلط باشد دستور ۲ اجرا می‌شود.

 **نمونه:** برنامه‌ای بنویسید که دو عدد از کاربر بگیرد و سپس آن‌ها را به ترتیب بزرگ به کوچک نمایش دهد.

پاسخ:

```
INPUT A , B
IF A > B THEN PRINT A , B ELSE PRINT B , A
```

اگر بخواهیم در صورت درست بودن شرط، چند دستور خاص و در صورت غلط بودن آن چند دستور خاص دیگر اجرا شوند، می‌توانیم از شکل زیر استفاده کنیم:

```

IF شرط THEN
    مجموعه دستورات ۱
ELSE
    مجموعه دستورات ۲
END IF

```

در این حالت در صورت درست بودن شرط، مجموعه دستورات ۱ و در صورت غلط بودن آن مجموعه دستورات ۲ اجرا می‌شود.

 **نمونه:** برنامه‌ای بنویسید که نمره‌ی یک دانش‌آموز را بگیرد و بگوید قبول است یا تجدید.

پاسخ:

```

INPUT A
IF A >= 10 THEN PRINT "GHABOOL" ELSE PRINT "TAJDID"

```

 **آیا می‌دانید؟**

برای ترکیب چند شرط می‌توان بین آن دو از دو کلمه **AND** به معنی « و » و **OR** به معنی « یا » استفاده کرد. دقت کنید که در بیسیک بر خلاف ریاضی، شرط‌های دوتایی (مثلاً به صورت $10 < A < 20$) نداریم.

 **نمونه:** برنامه‌ای بنویسید که سن کاربر را بپرسد و فقط اگر کاربر نوجوانی بین ۱۰ تا ۱۵ سال بود به او سلام کند!

پاسخ :

```
INPUT sen
IF sen > 10 AND sen < 15 THEN PRINT "SALAM NOJAVOON!"
```

 **نمونه:** در مثال قبل اگر کاربر نوجوان نباشد از او خداحافظی کنید !

پاسخ :

```
INPUT sen
IF sen < 10 OR sen > 15 THEN PRINT "khodahafez !!"
```

۴.۴. دستور پرش



دستوری که می‌خواهیم در این بخش آن را به شما یاد بدهیم، دستور خوبی نیست (!) و خیلی از برنامه نویسان بزرگ از آن استفاده نمی‌کنند. دلیل استفاده نکردن از آن هم این است که این دستور، دنبال کردن اجرای برنامه را برای ما سخت می‌کند. فرض کنید بنا به دلایلی می‌خواهید از یک خط برنامه به خط دیگری بپرید. برای این کار هم زبان بیسیک دستور ویژه‌ای به شکل زیر دارد:

شماره خط GOTO

وقتی رایانه به دستور بالا برسد، می‌رود به سراغ خطی از برنامه که شماره آن را جلوی دستور GOTO نوشته‌ایم.

 **توجه:** برای هر خط از برنامه که خواستید شماره بگذارید، کافی است در ابتدای آن خط از برنامه، شماره‌اش را بنویسید و بعد از یک فاصله، دستور آن خط را بنویسید.

 **نمونه:** برنامه‌ای بنویسید که از کاربر دو عدد به همراه حاصل ضرب آن‌ها را بپرسد. اگر حاصل ضرب درست بود بنویسد AFARIN و برنامه تمام شود. در غیر این صورت دوباره حاصل ضرب را بپرسد و آن قدر این کار را تکرار کند تا حاصل ضرب درست وارد شود.

پاسخ:

```

INPUT A , B
10 INPUT hasel_zarb
IF hasel_zarb = A * B THEN
    PRINT " AFARIN !"
    END
ELSE
    PRINT " ESHTEBAH BOOD !"
    GOTO 10
END IF

```

.....

۴.۵. توابع پر کاربرد

برای انجام بعضی از اعمال ریاضی در زبان بیسیک دستوراتی وجود دارد که در این بخش مهم‌ترین آن‌ها را یاد می‌گیریم.

۴.۵.۱. تابع قسمت صحیح

اگر بخواهید رایانه قسمت صحیح (عدد صحیح قبل از) هر عدد را برای شما حساب کند، کافی است از عبارت زیر استفاده کنید:

INT (عبارت)

دقت کنید که خط بالا دستور نیست، بلکه یک عدد است که آن را می‌توان نمایش داد (با دستور PRINT) یا در عبارت‌ها استفاده کرد.

 نمونه:

```
PRINT INT (7.4)
PRINT INT (3)
PRINT INT (-1.5)
LET A = INT (7.4)
LET A = INT (B)
```

پاسخ: خطوط اول تا سوم به ترتیب اعداد 7، 3 و 2- را نمایش می‌دهند (توجه کنید که عدد صحیح قبل از 1.5-، برابر 2- است). خط چهارم مقدار 7 را در A ذخیره می‌کند و خط پنجم قسمت صحیح عدد ذخیره شده در خانه حافظه B را در A ذخیره می‌کند.

 نمونه: برنامه‌ای بنویسید که یک عدد بگیرد و بگوید صحیح است یا اعشاری.

پاسخ: هر عددی قسمت صحیح‌اش با خودش برابر باشد، صحیح است و در غیر این صورت اعشاری است.

```
INPUT A
IF A = INT (A) THEN PRINT "SAHIH" ELSE PRINT "ASHARI"
```

۴.۵.۲. تابع تولید عدد تصادفی

برای تولید یک عدد تصادفی بین ۰ و ۱ از دستور RND استفاده می‌کنیم. رایانه هر وقت به کلمه RND بر بخورد، به جای آن یک عدد اتفاقی اعشاری بین ۰ و ۱ قرار می‌دهد. (امکان دارد خود صفر تولید شود، اما یک نه)

 نمونه:

```
PRINT RND
A = RND
```

خط اول، یک عدد اتفاقی نمایش داده و خط دوم یک عدد اتفاقی را در خانه حافظه A ذخیره می‌کند.

برای تولید یک عدد اتفاقی صحیح بین A و B (با امکان تولید خود A و B) می‌توان از دستور زیر استفاده کرد:

$$\text{INT}(\text{RND} * (\text{B} - \text{A} + 1)) + \text{A}$$

 **نمونه:** برنامه‌ای بنویسید که با اجرای آن رایانه مثل تاس، یک عدد اتفاقی صحیح بین ۱ تا ۶ (شامل خود ۱ و ۶) بنویسد.

پاسخ:

```
PRINT INT (RND * 6) + 1
```

 **توجه:** اگر برنامه بالا را چند دفعه پشت سر هم اجرا کنید، می‌بینید هر دفعه همان جواب قبلی را می‌نویسد. این به این دلیل است که شما تصادفی بودن عددها را فعال نکرده‌اید. برای فعال کردن دستور RND، در ابتدای هر برنامه‌ای که در آن از دستور RND استفاده شده، دستور زیر را عیناً بنویسید:

Randomize timer

۴. ۵. ۳. عملگر باقی‌مانده

برای بدست آوردن باقی‌مانده‌ی تقسیم A بر B می‌توانید از دستور زیر استفاده کنید:

$$\text{A MOD B}$$

 **نمونه:**

```
PRINT 24 MOD 7  
PRINT 6 MOD 11
```

خط اول عدد 3 و خط دوم عدد 6 را نمایش می‌دهد.

۴.۵.۴. عملگر خارج قسمت

برای به دست آوردن خارج قسمت صحیح تقسیم A بر B ، ضمن استفاده از تابع INT ، می‌توانید از دستور زیر هم استفاده کنید:

$$A \setminus B$$

 نمونه:

```
PRINT INT (34 / 7)
PRINT 34 \ 7
LET A = 32 \ 5
```

خطهای اول و دوم، هر دو عدد 4 را نمایش می‌دهند و خط دوم عدد 6 را درون A ذخیره می‌کند.

۴.۶. تمرین‌های برنامه نویسی

۱. برنامه‌ای بنویسید که رایانه ولتاژ دو سر یک لامپ و حداکثر ولتاژ قابل تحمل آن لامپ را بپرسد. سپس بنویسد لامپ می‌سوزد یا نه.

۲. برنامه‌ای بنویسید که با اجرای آن رایانه سه عدد از ما گرفته و میانگین آن‌ها را بنویسد. سپس روی صفحه بنویسد:

aya edame midahid?

1. bale
2. na

آنگاه یک عدد از ما بگیرد. اگر ۱ بود به ابتدای برنامه برود و اگر ۲ بود برنامه پایان پذیرد.

۳. برنامه‌ای بنویسید که رایانه یک عدد از ما بگیرد. اگر عدد برابر ۱۳۸۶ نبود، برنامه پایان پذیرد. اگر بود، چهار عدد بگیرد و میانگین آن‌ها را بنویسد.

۴. برنامه‌ای بنویسید که یک کلمه بپرسد. اگر کلمه SALAM نبود، برنامه پایان پذیرد. در غیر این صورت دو عدد گرفته، باقی‌مانده تقسیم اولی بر دومی را بنویسد.

۵. برنامه‌ای بنویسید که رایانه با گرفتن ساعت به ما صبح به خیر، ظهر به خیر، عصر به خیر و شب به خیر بگوید.

تا ۱۰: صبح به خیر

۱۰ تا ۱۴: ظهر به خیر

۱۴ تا ۱۹: عصر به خیر

۱۹ به بعد: شب به خیر

۶. برنامه‌ای بنویسید که قابلیت انجام کارهای زیر را داشته باشد:

(۱) محاسبه محیط و مساحت مستطیل

(۲) محاسبه محیط و مساحت دایره

(۳) محاسبه میانگین سه عدد

پس از اجرای برنامه باید ابتدا نوشته شود:

Kodam Ra Mikhahid?

1. Mostatil

2. Dayereh

3. Miangin

سپس رایانه یک عدد از کاربر بگیرد و بر مبنای آن به قسمت انتخاب شده رفته و آن را اجرا کند و بعد برنامه تمام شود.

۷. برنامه‌ای بنویسید که دو عدد گرفته و خارج قسمت صحیح و باقی‌مانده تقسیم اولی بر دومی را بدون استفاده از دستورات MOD و \ بنویسد.

۸. برنامه‌ای بنویسید که دو عدد گرفته و بگوید عدد اول بر عدد دوم بخش‌پذیر است یا نه.

۹. برنامه‌ای بنویسید که یک عدد گرفته و آن را گرد کرده و حاصل را نمایش دهد.

۱۰. برنامه‌ای بنویسید که یک عدد بی‌پرسد و بگوید زوج است یا فرد.

۱۱. برنامه‌ای بنویسید که یک عدد صحیح اتفاقی بین ۷ تا ۵۴۰ بنویسد. (امکان نوشتن خود ۷ و ۵۴۰ هم وجود داشته باشد)

۱۲. برنامه‌ای بنویسید که یک دانش‌آموز به کمک آن بتواند ضرب را تمرین کند. به این شکل که رایانه ابتدا دو عدد صحیح اتفاقی بین ۱۰ تا ۹۹ بنویسد. سپس حاصل ضرب آن دو را بی‌پرسد. اگر حاصل ضرب درست گفته شده بود، بنویسد AFARIN. اگر هم غلط بود، حاصل ضرب صحیح را بنویسد.

۱۳. کنار دستورات درست "ص" و کنار دستورات غلط "غ" بگذارید.

- ☐ INPUT Komput
- ☐ INPUT PRINT
- ☐ PRINT Shalgham
- ☐ IF A THEN PRINT B
- ☐ PRINT 2A , Salam
- ☐ IF A = 1 THEN INPUT B4
- ☐ IF A > B * 2 PRINT Salam
- ☐ PRINT A * 2 ; B + 4 , Salam
- ☐ IF A = "3" THEN PRINT "Salam"
- ☐ INPUT A * 2 , B + 4
- ☐ PRINT Sal ; am

- PRINT "GOTO IF "
- IF $A + B = 2^3$ THEN PRINT 2^3
- PRINT $\{s * 2^8 + 4\}$
- IF $A = 5$ THEN $A + 1$
- IF Folan = Folani THEN IF $A = 2$ THEN END

IF. خروجی برنامه زیر را بنویسید.

```
10 LET A = 3
20 LET B = A - 1
30 LET C = B * A
40 PRINT C
50 LET A = A + 1
60 LET B = B + 1
70 IF C < 40 THEN GOTO 20
```

