

---

---

# حل مسائل پایتون

## (حل ۱۰۰۰ برنامه - مرجع کامل)

---

---

### تألیف:

دکتر رمضان عباس نژادورزی  
مهندس یوسف عباس نژادورزی  
مهندس محمد نادعلی زاده چاری



فن آوری نوین

---

---

|                         |  |
|-------------------------|--|
| سرشناسه                 | : عباس نژاد ورزی، رمضان، ۱۳۴۸ -  |
| عنوان و نام پدیدآور     | : حل مسایل پایتون (حل ۱۰۰۰ برنامه - مرجع کامل) / تالیف رمضان عباس نژاد ورزی، یوسف عباس نژاد ورزی، محمد نادعلی زاده جاری.   |
| وضعیت ویراست            | : [ویراست ۳].  |
| مشخصات نشر              | : بابل: فناوری نوین، ۱۳۹۹.   |
| مشخصات ظاهری            | : ۵۹۶ ص.   |
| شابک                    | : ۱-۷۳۹۳۰۷-۶۲۲-۹۷۸   |
| وضعیت فهرست نویسی       | : فیبا   |
| یادداشت                 | : کتابنامه: ص. ۵۹۶.  |
| موضوع                   | : پایتون (زبان برنامه نویسی کامپیوتر) -- مسائل، تمرین ها و غیره<br>Python (Computer program language) -- Problems, exercises, etc<br>پایتون (زبان برنامه نویسی کامپیوتر)<br>Python (Computer program language) |
| شناسه افزوده            | : عباس نژاد ورزی، یوسف، ۱۳۶۴ -   |
| شناسه افزوده            | : نادعلی زاده جاری، محمد، ۱۳۶۴ -   |
| رده بندی کنگره          | : ۷۶/۷۳QA  |
| رده بندی دیویی          | : ۰۰۵/۱۳۳  |
| شماره کتابشناسی ملی     | : ۷۲۹۶۸۲۲  |
| اطلاعات رکورد کتابشناسی | : فیبا   |

[www.fanavarienovin.net](http://www.fanavarienovin.net)

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، کد پستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

### حل مسائل پایتون (حل ۱۰۰۰ برنامه - مرجع کامل)

تألیف: رمضان عباس نژاد ورزی، یوسف عباس نژاد ورزی، محمد نادعلی زاده جاری  
نوبت چاپ: چاپ سوم (ویراست سوم)  
سال چاپ: تابستان ۹۹  
شمارگان: ۲۰۰  
قیمت: ۱۰۵۰۰۰۰ تومان

نام چاپخانه و صحافی: دفتر فنی سورنا

شابک: ۹۷۸-۶۲۲-۷۳۹۳-۰۷-۱

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب مسجد منصور کاظم بیگی، طبقه اول

طراح جلد: کانون آکھی و تبلیغات آبان (احمد فرجی)

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

## فهرست مطالب

|  |     |
|--|-----|
| فصل اول: آشنایی با پایتون (دستورات ورودی و خروجی)..... | ۵   |
| فصل دوم: ساختار تصمیم و حلقه تکرار .....               | ۳۴  |
| فصل سوم: متدها در پایتون .....                         | ۱۷۳ |
| فصل چهارم: آرایه‌ها و NumPy .....                      | ۲۶۲ |
| فصل پنجم: رشته‌ها .....                                | ۳۸۰ |
| فصل ششم: لیست، چندتایی، دیکشنری .....                  | ۴۲۰ |
| فصل هفتم: کلاس‌ها و وراثت .....                        | ۴۶۵ |
| فصل هشتم: فایل‌های ورودی و خروجی .....                 | ۴۸۸ |
| پیوست الکترونیکی: تمرینات اضافی .....                  | ۵۰۹ |
| منابع: .....   | ۵۹۶ |

## مقدمه

پایتون یک زبان برنامه‌نویسی همه منظوره، سطح بالا، شیء‌گرا و مفسری است که توسط فان روسوم در سال ۱۹۹۱ در کشور هلند طراحی گردید. فلسفه ایجاد این زبان بر دو هدف اصلی خوانایی بالایی برنامه‌ها نوشته‌شده، کوتاهی و بازدهی نسبی بالای آن است. کلمات کلیدی این زبان به صورت حداقلی تهیه شده‌اند و در مقابل کتابخانه‌هایی که در اختیار کاربر است، بسیار وسیع هستند. کتاب حاضر اولین کتاب حل مسائل پایتون است که شامل ۸ فصل است. فصل‌های این کتاب عبارت‌اند از:

فصل اول، مفاهیم اولیه پایتون و دستورات ورودی و خروجی را با ۸۰ مثال آموزش می‌دهد.

فصل دوم، ساختارهای تکرار و تصمیم را با ۲۳۵ مثال کاربردی آموزش می‌دهد.

فصل سوم، تعریف متدها (توابع)، فراخوانی آن‌ها و استفاده از توابع کتابخانه‌ای را با ۱۰۷ مثال کاربردی آموزش می‌دهد.

فصل چهارم، آرایه‌ها یک‌بعدی، دوبعدی، چندبعدی و NUMPY را با ۸۱ مثال کاربردی آموزش می‌دهد.

فصل پنجم، رشته‌ها را با ۵۲ مثال برنامه‌نویسی آموزش داده است.

فصل ششم، چندیابی، مجموعه‌ها، لیست‌ها و دیکشنری‌های را با ۱۲۳ مثال برنامه‌نویسی آموزش داده است.

فصل هفتم، شیء‌گرایی، کلاس‌ها، وراثت، چندریختی و تعریف مجدد عملگرها را با ۱۹ مثال کاربردی آموزش داده است.

فصل هشتم، مفاهیم پایه‌ای ورودی و خروجی فایل‌های متنی و باینری را با ۴۰ مثال کاربردی آموزش داده است.

بالاخره، در پیوست الکترونیکی حدود ۲۹۰ مثال بیان گردیده و حل شده است.

از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهاد و انتقادات را به آدرس پست الکترونیک [fanavarienovin@gmail.com](mailto:fanavarienovin@gmail.com) ارسال نمایند.

در پایان امیدوارم این اثر مورد توجه جامعه انفورماتیک کشور، اساتید و دانشجویان عزیز قرار گیرد.

مؤلفین

[fanavarienovin@gmail.com](mailto:fanavarienovin@gmail.com)

## فصل

### ۱

# آشنایی با پایتون ( دستورات ورودی و خروجی )

۱. برنامه‌ای که قاعده و ارتفاع متوازی‌الاضلاع را خوانده، مساحت آن را نمایش می‌دهد (مساحت متوازی‌الاضلاع برابر با قاعده \* ارتفاع است).

```
base = float(input("length of base: "))
height = float(input('measurement of height: '))
area = base * height
print("Area is:", area)
```

دستور اول، با پیغام مناسبی قاعده را خوانده، با تابع float() به عدد اعشاری تبدیل می‌نماید و در متغیر base قرار می‌دهد، دستور دوم، با پیغام مناسب دیگری ارتفاع را خوانده، با متد float() آن را به عدد اعشاری تبدیل نموده، در متغیر height قرار می‌دهد، دستور سوم، قاعده (base) ضرب در ارتفاع (height) را در مساحت (area) قرار می‌دهد و دستور چهارم، با پیغام مناسبی مساحت را نمایش می‌دهد.

خروجی برنامه:

```
length of base: 14
measurement of height: 15.5
Area is: 217.0
```

۲. برنامه‌ای که شعاع و ارتفاع استوانه‌ای را خوانده، حجم و مساحت کل استوانه را محاسبه می‌کند. حجم و مساحت کل استوانه به صورت زیر محاسبه می‌شود:

$$\text{حجم استوانه} = \pi * (\text{شعاع})^2 * \text{ارتفاع}$$
$$(\text{شعاع})^2 * (\pi * 2) + \text{ارتفاع} * \text{شعاع} * \pi * 2 = \text{حجم مساحت کل}$$

```
pi = 22 / 7
height = float(input('Height of cylinder: '))
radius = float(input('radius of cylinder: '))
volume = pi * radius * radius * height
surArea = ((2 * pi * radius) * height) + (2 * (pi * radius ** 2))
print("Volume is: ", volume)
print("surface Area is: ", surArea)
```

دستور اول، ۲۲ تقسیم بر ۷ را در pi قرار می‌دهد، دستورات دوم و سوم، با پیغام‌های مناسب ارتفاع و شعاع استوانه را خوانده، به عدد اعشاری تبدیل می‌نمایند و به ترتیب در متغیرهای height و radius قرار می‌دهند، دستور سوم، حجم استوانه را با فرمول مشخص شده حساب کرده و در متغیر volume قرار می‌دهند و دستور چهارم، با فرمول بیان شده حجم مساحت کل استوانه را حساب کرده، در surArea قرار می‌دهد و دستورات پنجم و ششم، با پیغام‌های مناسب حجم استوانه و حجم مساحت کل استوانه را نمایش می‌دهند.

خروجی برنامه:

```
Height of cylinder: 20
radius of cylinder: 2
Volum is: 251.42857142857142
surface Area is: 276.57142857142856
```

۳. برنامه‌ای که شعاع کره‌ای را خوانده، مساحت و حجم کره را محاسبه می‌کند. مساحت و حجم کره برابر است با:

$$\text{مساحت کره} = 4 * \pi * (\text{شعاع})^2$$

$$V = \frac{4}{3} \pi r^3$$

```
pi=22/7
radius = float(input('Radius of sphere: '))
surArea = 4 * pi * radius **2
volume = (4/3) * (pi * radius ** 3)
print("Surface Area is: ", surArea)
print("Volume is: ", volume)
```

دستور اول مقدار pi را ۲۲ تقسیم بر ۷ تعیین می‌کند، دستور دوم، با پیغام مناسب، شعاع کره را خوانده، به عدد اعشاری تبدیل می‌نماید و در متغیر radius قرار می‌دهد، دستورات سوم و چهارم، با فرمول بیان شده، مساحت و حجم کره را حساب می‌کنند (\*\* عملگر توان است) و دستورات پنجم و ششم، با پیغام مناسب مساحت و حجم کره را نمایش می‌دهند.

خروجی برنامه:

```
Radius of sphere: 10
Surface Area is: 1257.142857142857
Volume is: 4190.47619047619
```

۴. برنامه‌ای که تعداد اضلاع و طول یک ضلع چندضلعی را خوانده، مساحت آن را با فرمول زیر محاسبه می‌کند:

$$A = \frac{s^2 \cdot n \cdot \tan(\pi/n)}{4}$$

```
from math import tan, pi
nSides = int(input("Input number of sides: "))
sLength = float(input("Input the length of a side: "))
pArea = nSides * (sLength ** 2) / (4 * tan(pi / nSides))
print("The area of the polygon is: ",pArea)
```

دستور اول، تابع tan و مقدار pi را از ماژول math به برنامه اضافه می‌کند، دستورات دوم و سوم، تعداد اضلاع و طول هر ضلع چندضلعی را خوانده، تعداد اضلاع چندضلعی را با تابع int() به عدد صحیح تبدیل کرده، در nSides قرار می‌دهد. طول هر ضلع چندضلعی را با تابع float() به عدد اعشاری تبدیل نموده، در sLength قرار می‌دهد، دستور چهارم با فرمول بیان شده مساحت چندضلعی را محاسبه می‌کند و در متغیر pArea قرار می‌دهد، دستور آخر، با پیغام مناسبی مساحت چندضلعی را نمایش می‌دهد.

خروجی برنامه:

```
Input number of sides: 12
Input the length of a side: 8
The area of the polygon is: 716.5537550532244
```

۵. برنامه‌ای که سرعت باد در ارتفاع ۱۰ متری را به متر در ثانیه (v) و دمای هوا به درجه سانتی‌گراد (T) را خوانده، شاخص وزش باد (wind chill) را محاسبه کند و نمایش می‌دهد:

$$wci = 13.12 + 0.6215 \cdot t - 11.37 \cdot v^{0.16} + 0.3965 \cdot t \cdot v^{0.16}$$

```
import math
v = float(input("Input wind speed in kilometers/hour: "))
t = float(input("Input air temperature in degrees Celsius: "))
wci = 13.12 + 0.6215*t - 11.37*math.pow(v, 0.16) + 0.3965*t*math.pow(v, 0.16)
print("The wind chill index is", int(round(wci, 0)))
```

دستور اول، ماژول math را به برنامه اضافه می‌کند، دستورات دوم و سوم، به ترتیب سرعت باد و دمای هوا را با پیغام‌های مناسب خوانده، به عدد اعشاری تبدیل می‌نمایند و در متغیرهای v و t قرار می‌دهند، دستور

## ۷ آشنایی با پایتون ( دستورات ورودی و خروجی )

چهارم، با فرمول بیان شده، شاخص وزش باد را حساب کرده، در متغیر wci قرار می‌دهد و دستور پنجم، شاخص باد را به عدد صحیح تبدیل می‌نماید و با پیغام مناسبی نمایش می‌دهد.

خروجی برنامه:

```
Input wind speed in kilometers/hour: 48
Input air temperature in degrees Celsius: 14
The wind chill index is 11
```

۶. برنامه‌ای که یک رشته و تعداد تکرار آن را خوانده، رشته را به تعداد عدد وارد شده تکرار می‌نماید و نمایش می‌دهد (هدف برنامه استفاده از عملگر \* برای تکرار رشته است).

```
s = input("Enter a string: ")
rep = int(input("Enter repeat: "))
print (s * rep)
```

| متغیر | هدف              |
|-------|------------------|
| s     | رشته ورودی       |
| rep   | تعداد تکرار رشته |

دستور اول، با پیغام مناسبی یک رشته را خوانده، در متغیر s قرار می‌دهد، دستور دوم، تعداد تکرار رشته را با پیغام مناسب دیگری خوانده، به عدد صحیح تبدیل می‌نماید و در متغیر rep قرار می‌دهد، دستور سوم، s\*rep را نمایش می‌دهد. در اینجا عملگر \*، s را به تعداد rep بار تکرار خواهد کرد.

خروجی برنامه:

```
Enter a string: Python
Enter repeat: 5
PythonPythonPythonPythonPython
```

۷. برنامه‌ای که دو عدد را خوانده، معادل مختلط آن دو عدد را نمایش می‌دهد (عدد اول بخش real و عدد دوم بخش imag می‌باشد). هدف این برنامه ایجاد اعداد مختلط و نمایش آن‌ها است.

```
a = int(input("Enter real part: "))
b = int(input("Enter image part: "))
complex1 = complex(a, b)
print (complex1)
```

| متغیر    | هدف                 |
|----------|---------------------|
| a        | بخش real            |
| b        | بخش imag            |
| complex1 | عدد مختلط تولید شده |

دستورات اول و دوم، با پیغام مناسب بخش real و imag را خوانده، به عدد صحیح تبدیل می‌نمایند و در متغیرهای a, b قرار می‌دهند، دستور سوم، با متد complex() با اعداد a, b یک متغیر با نوع مختلط به نام complex1 ایجاد می‌کند و دستور چهارم، مقدار complex1 را نمایش می‌دهد.

خروجی برنامه:

```
Enter real part: 12
Enter image part: -4
12 - 4j
```

۸. وزن یک مولکول آب  $3.0 \times 10^{-23}$  گرم و وزن یک لیتر آب در حدود ۹۵۰ گرم است. برنامه‌ای که وزن آب را برحسب لیتر از ورودی خوانده، تعداد مولکول‌های آن را محاسبه می‌کند.

```
w = input("Enter w: ")
w = int(w)
m = 3.0e-23
l = 950
tedad = (w * l) / m
print("Tedad = ", tedad)
```

| متغیر | هدف                                    |
|-------|--|
| m     | وزن یک مولکول                          |
| l     | وزن یک لیتر آب                         |
| w     | وزن آب به لیتر از ورودی خوانده می‌شود. |
| tedad | تعداد مولکول‌های آب                    |

## ۸ فصل اول

دستور اول، وزن مولکول را خوانده، در w قرار می‌دهد، دستور دوم، w را با متد `int()` به عدد صحیح تبدیل کرده، در w قرار می‌دهد، دستورات سوم و چهارم، متغیرهای m و l را به ترتیب با مقادیر  $10^{-23} \times 3,0$  و ۹۵۰ تعریف می‌کنند، دستور پنجم، با استفاده از فرمول بیان شده تعداد مولکول‌ها را محاسبه کرده، در `tedad` قرار می‌دهد و دستور ششم، با پیام مناسب `tedad` را نمایش می‌دهد.

خروجی برنامه:

```
Enter w: 134
Tedad = 4.2433333333333333e+27
```

۹. هر سال برابر با  $10^7 \times 3/156$  ثانیه است. برنامه‌ای که سن شما را دریافت کرده، به ثانیه تبدیل می‌کند.

```
ss = 3.156e7
age = int(input("Enter age:"))
second = age * ss;
print("Second = ", second)
```

| متغیر  | هدف                 |
|--------|---------------------|
| ss     | هر سال بر حسب ثانیه |
| age    | سن به سال           |
| second | سن بر حسب ثانیه     |

دستور اول، ss را برابر  $3,156 \times 10^7$  قرار می‌دهد، دستور سوم، با پیام مناسب سن را خوانده، به عدد تبدیل می‌نماید و در `age` قرار می‌دهد، دستور سوم، `age` ضرب در ss را محاسبه کرده، در `second` قرار می‌دهد و دستور چهارم، با پیام مناسب `second` را نمایش می‌دهد.

خروجی برنامه:

```
Enter age:48
Second = 1514880000.0
```

۱۰. برنامه‌ای که حقوق کارمندی را خوانده، بیمه، مالیات و دریافتی او را محاسبه کرده، نمایش می‌دهد. بیمه و مالیات به ترتیب ۷ و ۱۰ درصد حقوق می‌باشند.

```
a = float(input("Enter salary:"))
b = a * 0.07
m = a * 0.1
p = a - b - m
print("s = ", a, " b = ", b, " m = ", m, " p = ", p)
```

| متغیر | هدف     |
|-------|---------|
| s     | حقوق    |
| b     | بیمه    |
| m     | مالیات  |
| p     | دریافتی |

دستور اول، با پیام مناسب حقوق را خوانده، به عدد اعشاری تبدیل می‌کند و در `a` قرار می‌دهد، دستور دوم، `a` ضرب ۷ تقسیم بر  $100$  (یا  $0,07$ ) را محاسبه کرده، در `b` قرار می‌دهد، دستور سوم، `a` ضرب ۱۰ درصد را حساب کرده، در `m` قرار می‌دهد، دستور چهارم، حقوق خالص را برابر `a - b - m` قرار می‌دهد و دستور آخر، با پیام‌های مناسب `a`، `b`، `m`، `p` را نمایش می‌دهد.

خروجی برنامه:

```
Enter salary:10000000
s = 10000000.0 b = 700000.0000000001 m = 1000000.0 p = 8300000.0
```

۱۱. برنامه‌ای که قیمت کالایی را در سال قبل و جاری خوانده، نرخ تورم و قیمت سال بعد آن را چاپ می‌کند. نرخ تورم باید به صورت درصد (مثلاً ۶/۵٪) حساب گردد. (توضیح: برای محاسبه نرخ تورم و قیمت سال بعد به صورت زیر عمل کنید):



۹ آشنایی با پایتون ( دستورات ورودی و خروجی )

$$\text{قیمت سال قبل} - \text{قیمت سال جاری} = \text{نرخ تورم} \times \text{قیمت سال قبل}$$

$$\text{نرخ تورم} \times \text{قیمت سال جاری} + \text{قیمت سال قبل} = \text{قیمت سال بعد}$$

```
y1 = int(input("Enter price for first year:"))
y2 = int(input("Enter price for second year:"))
t = (float)(y2-y1) / y1
y3 = y1 + y2 * t;
print("Extera=%",t,"\t\tPrice next year = ",y3)
```

| متغیر          | هدف                         |
|----------------|-----------------------------|
| t              | نرخ تورم                    |
| y <sup>۱</sup> | قیمت کالا در سال قبل        |
| y <sup>۲</sup> | قیمت کالا در سال جاری       |
| y <sup>۳</sup> | قیمت تخمینی کالا در سال بعد |

دستورات اول و دوم، با پیغام‌های مناسب قیمت برای اولین و دومین سال را خوانده، به عدد صحیح تبدیل می‌نمایند و در y<sup>۱</sup> و y<sup>۲</sup> قرار می‌دهند، دستور سوم، با فرمول بیان شده، نرخ تورم را محاسبه کرده، در t قرار می‌دهد، دستور چهارم، قیمت کالا در سال بعد را با توجه به فرمول بیان شده محاسبه کرده، در y<sup>۳</sup> قرار می‌دهد و دستور پنجم، مقدار نرخ (t) و قیمت کالا در سال بعد (y<sup>۳</sup>) را با پیغام مناسب نمایش می‌دهد.

خروجی برنامه:

```
Enter price for first year:1000000
Enter price for second year:1200000
Extera=% 0.2          Price next year = 1240000.0
```

۱۲. مدیر شرکتی به هر متخصص خود ماهانه ۷۵۰۰۰۰ تومان حقوق پرداخت می‌کند. او می‌خواهد بداند که اگر ۱۳/۵ درصد به حقوق هر متخصص اضافه کند، سالانه چقدر به هزینه شرکت اضافه می‌شود. برنامه‌ای که تعداد متخصصین شرکت را خوانده، این کار را انجام می‌دهد.

```
pay = 750000
tedad = int(input("Enter tedad:"))
extera=pay * 12 * tedad * 13.5 / 100
print("Extera is ", extera)
```

| متغیر  | هدف                      |
|--------|--------------------------|
| pay    | ۷۵۰۰۰۰ (حقوق هر متخصص)   |
| tedad  | تعداد متخصصین شرکت       |
| extera | هزینه‌ای که اضافه می‌شود |

دستور اول، متغیر pay را با مقدار اولیه ۷۵۰۰۰۰ تعریف می‌کند، دستور دوم، تعداد متخصصین را با پیغام مناسب خوانده، با متد int() به عدد صحیح تبدیل کرده و در tedad قرار می‌دهد، دستور سوم، هزینه‌ای که اضافه می‌شود را با فرمول بیان شده محاسبه کرده، در extera قرار می‌دهد و دستور چهارم، مقدار extera را با پیغام مناسب نمایش می‌دهد.

خروجی برنامه:

```
Enter tedad:10
Extera is 12150000.0
```

۱۳. برنامه‌ای که عددی دورقمی را خوانده، مجموع ارقام و مغلوب آن را چاپ می‌کند.

```
num = int(input("Enter a number:"))
n1 = num % 10
n2 = num // 10
print("Reverse is ", n1, n2,"\t Sum is ", n1 + n2)
```

| متغیر          | هدف         |
|----------------|-------------|
| num            | عدد دریافتی |
| n <sup>۱</sup> | رقم یکان    |
| n <sup>۲</sup> | رقم دهگان   |

## ۱۰ فصل اول

دستور اول، با پیغام مناسب یک عدد صحیح را خوانده، در num قرار می‌دهد، دستور دوم، باقی‌مانده تقسیم صحیح num بر ۱۰ را محاسبه کرده (عملگر %). باقی‌مانده تقسیم صحیح را محاسبه می‌کند، در رقم یکان (n<sub>1</sub>) قرار می‌دهد، دستور سوم، تقسیم صحیح عدد num بر ۱۰ را حساب کرده (عملگر //) تقسیم صحیح عدد بر ۱۰ را حساب می‌کند، در رقم دهگان (n<sub>2</sub>) قرار می‌دهد، دستور چهارم، با پیغام مناسب مغلوب عدد و مجموع ارقام آن را نمایش می‌دهد.

خروجی برنامه:

Enter a number:59

Reverse is 9 5 Sum is 14

۱۴. برنامه‌ای که x را خوانده، بدون استفاده از عملگر ضرب، عبارت زیر را محاسبه می‌کند:

$$y = 31 * x - 17 * x + 5$$

(توضیح: در این برنامه، برای انجام ضرب از عملگر << شیفت به چپ استفاده می‌کنیم).

```
x = int(input("Enter x:"))
m = (x << 5) - x
n = -((x << 4) + x)
y = m + n + 5
print("y = ", y)
```

| متغیر | هدف             |
|-------|-----------------|
| m     | مقدار $x * 31$  |
| n     | مقدار $x * -17$ |
| x     | عدد دریافتی     |
| y     | نتیجه عبارت     |

دستور اول، با پیغام مناسب عددی را خوانده، در x قرار می‌دهد، دستور دوم، x را ۵ بار به سمت چپ شیفت می‌دهد تا در ۳۲ (۲<sup>۵</sup>) ضرب شود، سپس x را از آن کم می‌کند تا ۳۱ ضرب در x را محاسبه کند و نتیجه را در m قرار می‌دهد. دستور سوم، x را ۴ بار به سمت چپ شیفت می‌دهد تا ۱۶ ضرب در x حساب شود، سپس با x جمع می‌کند تا ۱۷ ضرب در x را محاسبه کند و نتیجه را در یک منفی ضرب می‌کند تا ۱۷x- محاسبه شود و آن را در n قرار می‌دهد، دستور چهارم، y را برابر m+n+5 قرار می‌دهد و دستور پنجم، با پیغام مناسب y را نمایش می‌دهد.

خروجی برنامه:

Enter x:17

y = 243

۱۵. برنامه‌ای که تاریخ و زمان فعلی را نمایش می‌دهد.

```
import datetime
now = datetime.datetime.now()
print("Current date and time:",now.strftime("%Y-%m-%d %H:%M:%S"))
```

دستور اول، ماژول datetime را به برنامه اضافه می‌کند، دستور دوم، با متد now()، تاریخ و زمان فعلی را در متغیر now قرار می‌دهد و دستور سوم، مقدار متغیر now را با پیغام مناسب نمایش می‌دهد.

خروجی برنامه:

Current date and time : 2017-06-10 20:22:32

۱۶. برنامه‌ای که عدد صحیح n را خوانده، حاصل عبارت n+nn+nnn را نمایش می‌دهد.

```
n = int(input("Input an integer : "))
n1 = int( "%s" % n )
n2 = int( "%s%s" % (n,n) )
n3 = int( "%s%s%s" % (a,n,n) )
print (n1, " + ", n2," +", n3, " = ", n1+ n2 +n3)
```

## ۱۱ آشنایی با پایتون ( دستورات ورودی و خروجی )

دستور اول، با پیغام مناسب عدد صحیح n را می خواند، دستور سوم، رشته n را به int تبدیل می کند و در n قرار می دهد، دستور سوم، رشته "nn" (یعنی اگر n، ۸ باشد "۸۸") را به عدد صحیح تبدیل می نماید و در n<sup>۲</sup> قرار می دهد، دستور چهارم، رشته "nnn" را به عدد صحیح تبدیل می کند و در n<sup>۳</sup> قرار می دهد، دستور چهارم، عبارت خواسته شده و مجموع آن ها را نمایش می دهد.

خروجی برنامه:

```
Input an integer : 8
8 + 88 + 888 = 984.
```

۱۷. برنامه ای که مبلغ فعلی، نرخ بهره و تعداد سال را خوانده و ارزش پول را برای چند سال آینده که از ورودی خوانده، محاسبه می کند.

```
amt = int(input("Enter amount:"))
rate = float(input("Enter rate:"))
years = int(input("Enter years:"))
future_value = amt*((1+(0.01*rate)) ** years)
print("Future_value is %12.0f" %(future_value))
```

دستور اول، با پیغام مناسب مبلغ فعلی را خوانده، در amt قرار می دهد، دستور دوم، نرخ سود را خوانده (به صورت عدد اعشاری) و در rate قرار می دهد، دستور سوم، با پیغام مناسب تعداد سال را خوانده، در years قرار می دهد، دستور چهارم، ابتدا، rate ضرب در ۰،۰۱ می کند تا درصد حساب شود، سپس آن را به علاوه ۱ می کند تا افزایش برای سال بعد را محاسبه کند، در ادامه، نتیجه را به years می رساند تا افزایش برای years سال های آینده را محاسبه کند و آن را ضرب در amt کرده تا ارزش پول را برای چند سال آینده را حساب نماید، در پایان، نتیجه را در متغیر future\_value قرار می دهد و دستور پنجم، future\_value را به صورت اعشاری با ۱۲ رقم با پیغام مناسب نمایش می دهد.

خروجی برنامه:

```
Enter amount:1000000
Enter rate:7.5
Enter years:10
Future_value is 2061032
```

۱۸. برنامه ای که مقدار x را از ورودی خوانده، حاصل عبارت مقابل را محاسبه می کند:

$$y = \frac{1}{x^2 + x + 1}$$

```
x = int(input("Enter x:"))
y = 1.0 / (x * x + x + 1)
print("y is ", y)
```

| متغیر | هدف         |
|-------|-------------|
| X     | ورودی       |
| y     | نتیجه عبارت |

دستور اول، با پیغام مناسب x را خوانده، دستور دوم، با فرمول بیان شده، y را محاسبه می کند و دستور سوم، با پیغام مناسب y را نمایش می دهد.

خروجی برنامه:

```
Enter x:10
y is 0.009009009009009009
```

۱۹. برنامه ای که دو عدد را از ورودی می خواند (عدد اول مقاومت یک مدار الکتریکی و عدد دوم جریان آن را نمایش می دهد). با فرمول مقابل ولتاژ را محاسبه کرده، نمایش می دهد:

جریان \* مقاومت = ولتاژ

## ۱۲ فصل اول

```
r = int(input("Enter R:"))
a = int(input("Enter A:"))
v = r * a
print("V is ", v)
```

| متغیر | هدف    |
|-------|--------|
| r     | مقاومت |
| a     | جریان  |
| v     | ولتاژ  |

این برنامه، ابتدا با یک پیام مقاومت (r) و جریان (a) را دریافت می‌نماید. در ادامه، ولتاژ را محاسبه نموده، نمایش می‌دهد. دستور اول با پیغام مناسب مقاومت را خوانده، در a قرار می‌دهد، دستور دوم، نیز با پیغام مناسب جریان را خوانده، در r قرار می‌دهد، دستور سوم، (جریان \* مقاومت) را حساب کرده، در v قرار می‌دهد و دستور آخر با پیغام مناسب ولتاژ (v) را نمایش می‌دهد.

خروجی برنامه:

```
Enter R:10
Enter A:20
V is 200
```

۲۰. مجموع مقاومت‌های R<sub>1</sub>، R<sub>2</sub> و R<sub>3</sub> که به‌طور موازی به هم متصل‌اند، با فرمول زیر محاسبه می‌شود:

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} = \frac{1}{R} = \frac{R_2 * R_3 + R_1 * R_3 + R_1 * R_2}{R_1 * R_2 * R_3}$$

برنامه‌ای که سه مقاومت را از ورودی خوانده، مقاومت کل را محاسبه می‌کند.

```
r1 = int(input("Enter R1:"))
r2 = int(input("Enter R2:"))
r3 = int(input("Enter R3:"))
r = (r1*r2*r3) / (r1 * r2 + r2 * r3 + r1 * r3)
print("R is ", r)
```

| متغیر          | هدف       |
|----------------|-----------|
| r <sub>1</sub> | مقاومت ۱  |
| r <sub>2</sub> | مقاومت ۲  |
| r <sub>3</sub> | مقاومت ۳  |
| r              | مقاومت کل |

این برنامه متغیرهای r<sub>1</sub>، r<sub>2</sub>، r<sub>3</sub> و r را با یک پیام می‌خواند. سپس مقاومت r را با فرمول بیان شده محاسبه کرده، نمایش می‌دهد.

خروجی برنامه:

```
Enter R1:10
Enter R2:20
Enter R3:30
R is 5.454545454545454
```

۲۱. برنامه‌ای که وزن کلایی را برحسب کیلوگرم دریافت می‌نماید و وزن آن را برحسب گرم نمایش می‌دهد. هر کیلوگرم برابر ۱۰۰۰ گرم است.

```
kg = int(input("Enter Kg:"))
g = kg * 1000
print("Weight(g) is ", g)
```

| متغیر | هدف           |
|-------|---------------|
| kg    | کیلوگرم (وزن) |
| g     | وزن به گرم    |

دستور اول، با پیغام مناسب وزن به کیلوگرم را خوانده، در kg قرار می‌دهد، دستور دوم، آن را در ۱۰۰۰ ضرب کرده تا به گرم تبدیل نماید و در متغیر g قرار می‌دهد، دستور سوم، با پیغام مناسب g را نمایش می‌دهد.

خروجی برنامه:

```
Enter Kg:20
Weight(g) is 20000
```

### آشنایی با پایتون ( دستورات ورودی و خروجی ) ۱۳

۲۲. اگر کارمندی به اندازه ۱۵ درصد حقوق ماه گذشته خود پاداش بگیرد، برنامه‌ای که حقوق کارمند را خوانده، پاداش او را محاسبه نموده، نمایش می‌دهد.

```
salary= int(input("Enter Salary:"))
reward = salary * 0.15
print("Reward is ",reward)
```

| متغیر  | هدف   |
|--------|-------|
| salary | حقوق  |
| reward | پاداش |

دستور اول، با پیغام مناسب حقوق را خوانده، در salary قرار می‌دهد، دستور دوم، salary را ضرب ۰,۱۵، (۱۵ درصد) می‌کند تا پاداش را حساب کرده، در reward قرار می‌دهد و دستور سوم، با پیغام مناسب reward را نمایش می‌دهد.

خروجی برنامه:

```
Enter Salary:10000000
Reward is 1500000.0
```

۲۳. برنامه‌ای که ارتفاع و قاعده مثلث را از ورودی خوانده، مساحت آن را محاسبه کرده، به خروجی می‌برد.

قاعده \* ارتفاع × ۰,۵ = مساحت مثلث

```
x = int(input("Enter x:"))
y = int(input("Enter y:"))
s = 1 / 2 * x * y
print("Area is " , s)
```

| متغیر | هدف    |
|-------|--------|
| x     | ارتفاع |
| y     | قاعده  |
| s     | مساحت  |

دستورات اول و دوم، با پیغام‌های مناسب ارتفاع و قاعده مثلث را خوانده، در متغیرهای x و y قرار می‌دهند، دستور سوم، ارتفاع (x) ضرب در قاعده (y) تقسیم بر ۲ را حساب کرده، در مساحت (s) قرار می‌دهد و در پایان، s را با پیغام مناسب نمایش می‌دهد.

خروجی برنامه:

```
Enter x:10
Enter y:20
Area is 100.0
```

۲۴. در شرکتی، سالانه ۱۵۰ خودکار، ۵۰ بسته کاغذ A۴ مصرف می‌شود. در پایان سال این شرکت می‌خواهد بداند در سال آینده چقدر برای این بخش از تجهیزات اداری، افزایش هزینه دارد. برنامه‌ای که قیمت این اقلام را در امسال از ورودی دریافت می‌نماید، نرخ تورم در سال آینده را نیز می‌گیرد، هزینه اضافه‌شده شرکت را در این بخش محاسبه می‌کند و به خروجی می‌برد. تورم به صورت درصد وارد می‌گردد که برنامه باید آن را به مقدار اعشاری تبدیل کند. به‌عنوان مثال، اگر تورم را ۵,۶ وارد کنیم، برنامه باید از ۰/۰۵۶ استفاده کند.

```
a4 = int(input("Enter a4 price:"))
pen = int(input("Enter pen price:"))
t = int(input("Enter t:"))
coste = 50*pen*t/ 100 + 150 * a4 * t/100
print("Extra cost is ",coste)
```

| متغیر | هدف             |
|-------|-----------------|
| a۴    | قیمت یک بسته A۴ |
| pen   | قیمت خودکار     |
| t     | تورم            |
| coste | هزینه اضافه‌شده |

دستورات اول تا سوم، با پیغام مناسب قیمت کاغذ، خودکار و تورم را خوانده، در متغیرهای a۴ و pen و t قرار می‌دهند، دستور چهارم، هزینه اضافه‌شده را محاسبه می‌کند و در coste قرار می‌دهد و دستور آخر با پیغام مناسب مقدار coste را نمایش می‌دهد.

خروجی برنامه:

```
Enter a4 price:10000
Enter pen price:1000
Enter t:12
Extra cost is 186000.0
```

۲۵. یک دوچرخه سوار با سرعت  $x$  کیلومتر بر ساعت شروع به حرکت می کند و پس از  $n$  دقیقه سرعت آن به  $k$  کیلومتر در ساعت می رسد. برنامه ای که با استفاده از فرمول زیر شتاب او را محاسبه کرده، نمایش می دهد:

$$\text{شتاب} = \frac{(k(\text{سرعت نهایی}) - x(\text{سرعت اولیه})) * 60}{n(\text{زمان})}$$

```
k = int(input("Enter k:"))
x = int(input("Enter x:"))
n = int(input("Enter n:"))
acceleration = (k - x) * 60 / n
print("acceleration is ", acceleration)
```

| هدف        | متغیر        |
|------------|--------------|
| سرعت نهایی | k            |
| سرعت اولیه | x            |
| زمان       | n            |
| شتاب       | acceleration |

دستورات اول تا سوم، با پیغام های مناسب سرعت نهایی، سرعت اولیه و زمان را خوانده، به ترتیب در متغیرهای  $k$ ،  $x$  و  $n$  قرار می دهند، دستور چهارم، با فرمول مشخص شده، شتاب را حساب کرده، در متغیر acceleration قرار می دهد و دستور آخر، شتاب را با پیغام مناسب نمایش می دهد.

خروجی برنامه:

```
Enter k:15
Enter x:15
Enter n:5
acceleration is 60
```

۲۶. برنامه ای که مصرف (۱۰۰ کیلومتر به لیتر) و گنجایش باک بنزین (لیتر) اتومبیلی را از ورودی می خواند و تعیین می کند که با یک باک بنزین چه مسافتی را می تواند بپیماید.

```
consumption=int(input("Enter Consumption :"))
capacity = int(input("Enter Capacity:"))
distance = capacity / consumption * 100
print("Distance is ",distance)
```

| هدف   | متغیر       |
|-------|-------------|
| ظرفیت | capacity    |
| فاصله | distance    |
| مصرف  | consumption |

دستور اول و دوم، با پیغام مناسب مصرف و ظرفیت را خوانده، به ترتیب در متغیرهای consumption و Capacity قرار می دهند، دستور سوم، ظرفیت را تقسیم بر مصرف کرده، سپس ضرب در ۱۰۰ می کند تا فاصله به دست آید و آن را در متغیر distance قرار می دهد و دستور چهارم، با پیغام مناسب distance را نمایش می دهد.

خروجی برنامه:

```
Enter Consumption :10
Enter Capacity:60
Distance is 600
```

۲۷. برنامه ای که دو عدد را از کاربر دریافت می کند و حاصل جمع، ضرب، تقسیم، تفریق و باقی مانده تقسیم صحیح آن ها را چاپ می کند.

| هدف     | متغیر |
|---------|-------|
| عدد اول | a     |

## آشنایی با پایتون ( دستورات ورودی و خروجی ) ۱۵

```
a = int (input("Enter a:"))
b = int (input("Enter b:"))
print(a, " + ", b," = ", a + b)
print(a, " - ", b," = ", a - b)
print(a, " * ", b," = ", a * b)
print(a, " / ", b," = ", a / b)
print(a, " % ", b," = ", a % b)
```

|         |   |
|---------|---|
| عدد دوم | b |
|---------|---|

دستورات اول و دوم، دو عدد را خوانده، در متغیرهای a و b قرار می‌دهند، دستور سوم، a، "+"، b، "="، جمع a و b را نمایش می‌دهد، دستور چهارم، مانند دستور سوم است، با این تفاوت که تفریق b از a را نمایش می‌دهد، دستورات پنجم تا هفتم، به ترتیب حاصل ضرب a و b، تقسیم a بر b و باقی‌مانده تقسیم صحیح a بر b را نمایش می‌دهند.

**خروجی برنامه:**

```
Enter a:10
Enter b:6
6 + 10 = 16
10 - 6 = 4
10 * 6 = 60
10 / 6 = 1.6666666666666667
10 % 6 = 4
```

**۲۸. برنامه‌ای که تعیین می‌کند آیا پوسنه اجرا ۳۲ بیتی است یا ۶۴ بیتی .**

```
import struct
print(struct.calcsize("P") * 8)
```

دستور اول، ماژول struct را به برنامه اضافه می‌کند، دستور دوم، با استفاده از متد calcsize("P")، اندازه پردازنده را به بایت محاسبه می‌کند، سپس آن را در ۸ ضرب می‌کند تا اندازه پردازنده به بیت حساب شود و آن را نمایش می‌دهد.

**خروجی برنامه:**

64

**۲۹. برنامه‌ای که نام سیستم‌عامل، اطلاعات پلت فرم release را نمایش می‌دهد.**

```
import platform
import os
print("Name is ", os.name)
print("Platform system is ", platform.system())
print("Platform release is ", platform.release())
```

دستورات اول و دوم به ترتیب ماژول‌های platform و os را به برنامه اضافه می‌کنند، دستور سوم، نام سیستم‌عامل را با پیام مناسب نمایش می‌دهد، دستور چهارم، نوع سیستم‌عامل را با پیام مناسب نمایش می‌دهد و دستور پنجم، نسخه سیستم‌عامل را با پیام مناسب نمایش خواهد داد.

**خروجی برنامه:**

```
Name is nt
Platform system is Windows
Platform release is 7
```

**۳۰. برنامه‌ای که نام سایت بسته‌های پایتون را نمایش می‌دهد.**

```
import site
print("Site package is ", site.getsitepackages())
```

دستور اول، ماژول site را به برنامه اضافه می‌کند، دستور دوم، با فراخوانی متد getsitepackages()، نام سایت سیستم پایتون را برگردانده و با پیام مناسب نمایش می‌دهد.

خروجی برنامه:

```
Site package is
['C:\\Users\\Fansno\\AppData\\Local\\Programs\\Python\\Python35',
'C:\\Users\\Fansno\\AppData\\Local\\Programs\\Python\\Python35\\lib\\site-
packages']
```

۳۱. برنامه‌ای که یک برنامه اجرایی خارجی را در پایتون اجرا می‌کند.

```
from subprocess import call
call(["calc.exe"])
```

دستور اول، متد call از بسته subprocess را به برنامه اضافه می‌کند، دستور دوم، با متد call() برنامه ماشین حساب (calc.exe) را اجرا می‌نماید.

۳۲. با استفاده از امکاناتی که در این فصل دیدید برنامه‌ای که توان ۲، توان ۳، و عدد ۰ تا ۱۰ را به صورت زیر جدول بندی می‌نماید؟ (توضیح: همان طور که مشاهده نمودید، تعداد خطوط این برنامه زیاد شده است که با بیان حلقه‌های تکرار در فصل بعد، تعداد دستورات این قبیل برنامه‌ها کاهش خواهد یافت).

```
print("Number\tSquare\tCube")
print(0, " ", "0 * 0", " ", "0 * 0 * 0")
print(1, " ", "1 * 1", " ", "1 * 1 * 1")
print(2, " ", "2 * 2", " ", "2 * 2 * 2")
print(3, " ", "3 * 3", " ", "3 * 3 * 3")
print(4, " ", "4 * 4", " ", "4 * 4 * 4")
print(5, " ", "5 * 5", " ", "5 * 5 * 5")
print(6, " ", "6 * 6", " ", "6 * 6 * 6")
print(7, " ", "7 * 7", " ", "7 * 7 * 7")
print(8, " ", "8 * 8", " ", "8 * 8 * 8")
print(9, " ", "9 * 9", " ", "9 * 9 * 9")
print(10, " ", "10*10", " ", "10*10* 10")
```

دستور اول، یک عنوان را نمایش می‌دهد. دستور دوم، توان ۲ صفر، توان ۳ صفر و توان ۳ صفر را نمایش می‌دهد، دستور سوم، ۱، توان ۲ یک و توان ۳ یک را نمایش می‌دهد و همین‌طور ادامه می‌یابد تا دستور آخر، ۱۰، توان ۲ ده و توان ۳ ده را نمایش می‌دهد.

خروجی برنامه:

| Number | Square | Cube |
|--------|--------|------|
| 0      | 0      | 0    |
| 1      | 1      | 1    |
| 2      | 4      | 8    |
| 3      | 9      | 27   |
| 4      | 16     | 64   |
| 5      | 25     | 125  |
| 6      | 36     | 216  |
| 7      | 49     | 343  |
| 8      | 64     | 512  |
| 9      | 81     | 729  |
| 10     | 100    | 1000 |

۳۳. برنامه‌ای که یک عدد صحیح پنج رقمی را خوانده، ارقام آن را جدا می‌کند و هر یک از ارقام را با سه فاصله بین آن‌ها چاپ می‌کند. به عنوان مثال، اگر کاربر عدد ۴۲۳۳۹ را وارد کند، خروجی به صورت زیر باشد:

Result is 4 2 3 3 9

| متغیر | هدف          |
|-------|--------------|
| n     | عدد پنج رقمی |
| temp  | متغیر کمکی   |
| r۱    | رقم اول      |



## آشنایی با پایتون ( دستورات ورودی و خروجی ) ۱۷

```
n=int(input("Enter a number between 10000 to 99999:"))
temp = n
r5 = temp % 10
temp = temp // 10
r4 = temp % 10
temp = temp // 10
r3 = temp % 10
temp = temp // 10
r2 = temp % 10
temp = temp // 10
r1 = temp % 10
temp = temp // 10
print("Result is",r1," ",r2," ",r3," ",r4," ",r5)
```

|           |   |
|-----------|---|
| رقم دوم   | ۲ |
| رقم سوم   | ۳ |
| رقم چهارم | ۴ |
| رقم پنجم  | ۵ |

دستور اول، یک عدد بین ۱۰۰۰۰ تا ۹۹۹۹۹ خوانده، در n قرار می‌دهد، دستور دوم، n را در temp قرار می‌دهد تا عدد n از دست نرود، دستور سوم، رقم یکان temp (از طریق عملگر %) را در r5 قرار می‌دهد، دستور چهارم، رقم یکان temp را حذف کرده (عملگر // تقسیم صحیح را انجام می‌دهد)، در temp قرار می‌دهد، دستور پنجم، رقم یکان temp (یعنی همان رقم دهگان n) را در r4 قرار می‌دهد، دستور ششم، رقم یکان temp را حذف می‌کند، و همین روند ادامه می‌یابد تا ارقام عدد را جدا کند. دستور آخر، رقم یکان، دهگان، صدگان، هزارگان و ده هزارگان را نمایش می‌دهد (یعنی، ارقام عدد برعکس نمایش داده می‌شود).

خروجی برنامه:

```
Enter a number between 10000 to 99999:67543
Result is 6 7 5 4 3
```

۳۴. برنامه‌ای که دو مقدار را خوانده، شماره شناسایی این اشیاء را نمایش می‌دهد.

```
a = int(input("Enter a: "))
b = input("Enter b: ")
print("ID a is " , id(a))
print("ID b is " , id(b))
```

| متغیر | هدف             |
|-------|-----------------|
| a     | مقدار ورودی اول |
| b     | مقدار ورودی دوم |

دستورات اول و دوم، به ترتیب یک عدد و یک رشته را خوانده، در متغیرهای a و b قرار می‌دهند، دستورات سوم و چهارم، با تابع id()، شماره شناسایی a و b را با پیغام مناسب نمایش می‌دهند.

خروجی برنامه:

```
Enter a: 10
Enter b: Fanavarienovin
ID a is 1498607696
ID b is 58207472
```

۳۵. هر لیتر معادل ۰٫۲۶۴۱۷۹ گالن است. برنامه‌ای که میزان بنزین مصرفی اتومبیل کاربر را بر حسب لیتر و مسافت طی شده آن را به مایل گرفته، سپس مصرف به ازای هر مایل به گالن را نمایش می‌دهد.

```
liter = int(input("Enter liter:"))
mile = int(input("Enter mile:"))
result = (mile / liter) * 0.264179
print("miles / gallons is ", result)
```

| متغیر  | هدف                    |
|--------|------------------------|
| liter  | مصرف به لیتر           |
| mile   | مسافت به مایل          |
| result | نسبت مصرف مایل به گالن |

دستورات اول و دوم، به ترتیب مصرف به لیتر و مسافت طی شده را خوانده، در متغیرهای liter و mile قرار می‌دهند، دستور سوم، با فرمول، نسبت مصرف مایل به گالن را حساب کرده، در result قرار می‌دهد و دستور چهارم، با پیغام مناسب result را نمایش می‌دهد.

خروجی برنامه:

```
Enter liter:10
Enter mile:100
miles / gallons is 2.64179
```

۳۶. برنامه‌ای که عددی را خوانده، مربع و مکعب آن را نمایش می‌دهد.

```
x = int(input("Enter x:"))
print(x, " ^ 2 = ", x ** 2, "\n", x, " ^ 3 = ", x ** 3)
```

| متغیر | هدف            |
|-------|----------------|
| X     | عدد خوانده شده |

دستور اول، عدد را خوانده، در X قرار می‌دهد و دستور دوم با پیغام مناسب توان ۲ و توان ۳ آن عدد را نمایش می‌دهد.

خروجی برنامه:

```
10 ^ 2 = 100
10 ^ 3 = 1000
```

۳۷. برنامه‌ای که دو ضلع موازی و ارتفاع (a، b و h) دوزنقه را دریافت کرده، مساحت آن را با فرمول زیر محاسبه می‌کند:

$$s = h * (a + b) / 2$$

```
a = int(input("Enter a:"))
b = int(input("Enter b:"))
h = int(input("Enter h:"))
s = h * (a + b) / 2
print("S is ", s)
```

| متغیر | هدف     |
|-------|---------|
| a     | ضلع اول |
| b     | ضلع دوم |
| h     | ارتفاع  |
| s     | مساحت   |

دستورات اول و دوم، با پیغام‌های مناسب به ترتیب دو ضلع موازی و ارتفاع دوزنقه را خوانده، در متغیرهای a و b و h قرار می‌دهند، دستور چهارم، با فرمول بیان شده، مساحت دوزنقه را حساب کرده، در s قرار می‌دهد و دستور پنجم، مساحت دوزنقه (s) را با پیغام مناسب نمایش می‌دهد.

خروجی برنامه:

```
Enter a:10
Enter b:12
Enter h:13
S is 143.0
```

۳۸. برنامه‌ای که شعاع دایره را خوانده، محیط، مساحت و قطر آن را نمایش می‌دهد (محیط، مساحت و قطر دایره به صورت زیر حساب می‌شوند):

$$\text{شعاع} * 2 = \text{قطر}$$

$$\pi * \text{شعاع} * 2 = \text{محیط دایره}$$

$$\pi * (\text{شعاع})^2 = \text{مساحت دایره}$$

```
PI = 3.1415
r = int(input("Enter r:"))
n = 2 * r
p = n * PI;
a = PI * r * r;
print(" n = ", n, "\n p = ", p, "\n a = ", a)
```

| متغیر   | هدف   |
|---------|---|
| ثابت pi | مقدار $\pi$ یعنی ۳,۱۴۱۵۹ را نگهداری می‌کند. |
| r       | شعاع خوانده شده را نگهداری می‌کند           |
| a       | مساحت دایره را نگهداری می‌کند.              |
| p       | محیط دایره را نگهداری می‌کند.               |
| n       | قطر دایره را نگهداری می‌کند.                |

## آشنایی با پایتون ( دستورات ورودی و خروجی ) ۱۹

دستور اول، مقدار pi را برابر ۳,۱۴۱۵ قرار می دهد، دستور دوم، با پیغام مناسب شعاع دایره را خوانده، در r قرار می دهد، دستور سوم، ۲ ضرب در r (همان قطر) را در n قرار می دهد، دستور چهارم، n (قطر) ضرب در pi را محاسبه کرده، در محیط (p) قرار می دهد، دستور پنجم، pi ضرب در r ضرب در r را در مساحت (a) قرار می دهد و دستور آخر، اطلاعات خواسته شده را نمایش می دهد.

### خروجی برنامه:

```
Enter r:20
n = 40
p = 125.66000000000001
a = 1256.6000000000001
```

۳۹. برنامه ای که عددی را به سال خوانده، تعیین می کند چند ماه، چند روز و چند ثانیه است.

سال \* ۳۶۵,۲۵ = روز  
روز / ۳۰ = ماه  
۶۰ \* ۶۰ \* ۲۴ \* روز = ثانیه

```
year = int(input("Enter year:"))
day = 365.25 * year;
print("Day is ", day)
month = day // 30;
print("Month is ", month)
second = day * 24 * 60 * 60
print("Second is ", second)
```

| متغیر  | هدف                    |
|--------|------------------------|
| year   | عدد ورودی به سال       |
| day    | عدد تبدیل شده به روز   |
| month  | عدد تبدیل شده به ماه   |
| second | عدد تبدیل شده به ثانیه |

دستور اول، با پیغام مناسب سال را خوانده، در متغیر year قرار می دهد، دستور دوم، با فرمول بیان شده روز را حساب کرده، در day قرار می دهد، دستور سوم، با پیغام مناسب day را نمایش می دهد، دستور چهارم، را بر ۳۰ به صورت صحیح تقسیم می کند (عملگر //) و در month قرار می دهد، دستور پنجم، month را با پیغام مناسبی نمایش می دهد، دستور ششم، day را در ۲۴ (هر روز ۲۴ ساعت) در ۶۰ (هر ساعت ۶۰ دقیقه) و در ۶۰ (هر دقیقه ۶۰ ثانیه) ضرب کرده، در متغیر second قرار می دهد و دستور، آخر مقدار second را با پیغام مناسب نمایش می دهد.

### خروجی برنامه:

```
Enter year:20
Day is 7305.0
Month is 243.0
Second is 63115200.0
```

۴۰. برنامه ای که m و n را خوانده، حاصل عبارات زیر را محاسبه کند:

$$a = m^2 - n^2$$

$$b = 2 * m * n$$

$$c = m^2 + n^2$$

```
m = int(input("Enter m:"))
n = int(input("Enter n:"))
a = m ** 2 - n ** 2
b = 2 * m * n
c = m ** 2 + n ** 2
print("a is ", a)
print("b is ", b)
print("c is ", c)
```

## ۲۰ فصل اول

دستورات اول و دوم، با پیغام مناسب m و n را خوانده، دستورات سوم تا پنجم، با فرمول‌های بیان‌شده a و b و c را محاسبه می‌کنند و سه دستور آخر، مقادیر a و b و c را با پیغام‌های مناسب نمایش می‌دهند.

خروجی برنامه:

```
Enter m:20
Enter m:17
a is 111
b is 680
c is 689
```

۴۱. برنامه‌ای که دو عدد را خوانده، در متغیرهای a و b قرار می‌دهد، سپس حاصل جمع، تفریق، ضرب، تقسیم،  $a^2+b^2$  و  $a^3+b^3$  را نمایش می‌دهد.

```
a = int(input("Enter a:"))
b = int(input("Enter b:"))
print("a + b = ",a + b)
print("a - b = ",a - b)
print(" a * b = ",a * b)
print("a / b = ",a / b)
print("a ^ 2 + b ^ 2 = ",a * a + b * b)
print("a ^ 3 + b ^ 3 = ",a * a * a + b * b * b)
```

دستورات اول و دوم، با پیغام‌های مناسب a و b را خوانده، دستور سوم، با پیغام مناسب a+b را نمایش می‌دهد، دستور چهارم، با پیغام مناسب a-b را نمایش خواهد داد، دستورات پنجم تا هشتم به ترتیب با پیغام‌های مناسب  $a^2+b^2$ ،  $a^3+b^3$ ،  $a/b$ ،  $a * b$  را نمایش می‌دهند.

خروجی برنامه:

```
Enter a:10
Enter b7
a + b = 17
a - b = 3
a * b = 70
a / b = 1.4285714285714286
a ^ 2 + b ^ 2 = 149
a ^ 3 + b ^ 3 = 1343
```

۴۲. برنامه‌ای که نام و مسیر فایل فعلی در حال اجرا را نمایش می‌دهد.

```
import os
print("Current File Name : ",os.path.realpath(__file__))
```

دستور اول، ماژول os را به برنامه اضافه می‌کند، دستور دوم، با فراخوانی متد `realpath()` مسیر فایل فعلی (که در متغیر `__file__` قرار دارد) را با پیغام مناسب نمایش می‌دهد.

خروجی برنامه:

```
Current File Name : D:\BookCSharp\python-book\program\site\py\15.py
```

۴۳. فرض کنید هر صفحه کتاب ۳۰ سطر (خط) است و هر خط ۸۰ بایت حافظه اشغال می‌کند. برنامه‌ای که تعداد صفحات یک کتاب و ظرفیت یک فلش را به گیگابایت خوانده، تعیین می‌کند که این فلش چند تا از این کتاب‌ها را می‌تواند در خودش ذخیره کند (گیگابایت برابر با  $1024^3$  بایت است).

| متغیر  | هدف  |
|--------|--|
| pages  | تعداد صفحات کتاب که از ورودی خوانده می‌شود.      |
| Gb     | ظرفیت فلش به گیگابایت که از ورودی خوانده می‌شود. |
| number | تعداد کتاب‌هایی که این فلش می‌تواند ذخیره کند.   |

## آشنایی با پایتون (دستورات ورودی و خروجی) ۲۱

bookBytes تعداد بایت‌هایی که این کتاب نیاز دارد.

```
pages = int(input("Enter pages:"))
Gb = int(input("Enter size of memory :"))
bookBytes = 80 * 30 * pages;
number = 1024 * 1024 * 1024 * Gb / bookBytes;
print("Number of book is ", number)
```

دستورات اول و دوم، به ترتیب با پیغام‌های مناسب تعداد صفحات کتاب و ظرفیت فلش را خوانده، در متغیرهای pages و Gb قرار می‌دهند، دستور سوم، تعداد کاراکترهای کتاب را حساب می‌کند (هر سطر ۸۰ کاراکتر و هر صفحه ۳۰ خط است)، در bookBytes قرار می‌دهد، دستور چهارم، ابتدا Gb را به بایت تبدیل کرده (یعنی، Gb را در  $1024^3$  ضرب می‌کند تا به بایت تبدیل شود) و بر تعداد بایت‌های یک کتاب (bookBytes) تقسیم می‌نماید تا تعداد کتاب‌ها به دست آید و آن را در number قرار می‌دهد و در پایان، number را با پیغام مناسب نمایش می‌دهد.

خروجی برنامه:

```
Enter pages:1096
Enter size of memory :16
Number of book is 6531.276301703163
```

۴۴. برنامه‌ای که تعداد پردازنده‌های رایانه را نمایش می‌دهد.

```
import multiprocessing
print("Count cpu is ", multiprocessing.cpucount())
```

دستور اول، ماژول multiprocessing را به برنامه اضافه می‌کند و دستور دوم، با پیغام مناسب و فراخوانی متد multiprocessing.cpucount() (تعداد cpu ها را برمی‌گرداند) را نمایش می‌دهد.

خروجی برنامه:

```
Count cpu is 2
```

۴۵. برنامه‌ای که تعداد کالا و قیمت هر کالا را خوانده، مبلغ فروش را نمایش می‌دهد (مبلغ فروش برابر با تعداد کالا \* قیمت کالا است).

```
count = int(input("Enter Count:"))
price = int(input("Enter Price:"))
sell = count * price
print("Sells is ", sell)
```

| متغیر | هدف        |
|-------|------------|
| count | تعداد کالا |
| price | قیمت کالا  |
| sell  | مبلغ کالا  |

دستورات اول و دوم، با پیغام مناسب تعداد کالا و قیمت آن را خوانده، در متغیرهای count و price قرار می‌دهند، دستور سوم،  $price * count$  (مبلغ کالا) را در sell قرار داده و دستور چهارم، با پیغام مناسب sell را نمایش می‌دهد.

خروجی برنامه:

```
Enter Count:10
Enter Price:675000
Sells is 6750000
```

۴۶. برنامه‌ای که سن‌تان را به سال خوانده، مشخص کند چند دقیقه زندگی کرده‌اید (هر سال ۳۶۵,۲۵ روز است و هر روز  $24 \times 60$  دقیقه می‌باشد).

| هدف                                      | متغیر  |
|--|--------|
| سن تان به سال که از ورودی خوانده می شود. | year   |
| سن تان که به دقیقه تبدیل شده است.        | minute |

```
year = int(input("Enter year:"))
minute = year * 365.25 * 24 * 60
print("Minute is ",minute)
```

دستور اول، با پیغام مناسب سن تان را به سال خوانده، در year قرار می دهد، دستور دوم، سن (year) را در ۳۶۵٫۲۵ ضرب می کند تا تعداد روز را به دست آورد. نتیجه این عملیات را در ۲۴ ضرب می کند تا تعداد ساعت را به دست آورد و سپس نتیجه را در ۶۰ ضرب می کند تا تعداد دقیقه را محاسبه نماید، و نتیجه را در minute قرار می دهد و دستور سوم، دقیقه را با پیغام مناسب نمایش می دهد.

خروجی برنامه:

```
Enter year:48
Minute is 25246080.0
```

۴۷. برنامه ای که مقداری را به مایل و فوت دریافت کرده، به متر و کیلومتر تبدیل می کند و نمایش می دهد (هر فوت ۰٫۳۰۴۸ متر و هر مایل ۱۶۰۹٫۳۵ متر است).

| هدف   | متغیر      |
|---|------------|
| ثابت نرخ تبدیل مایل به متر                                    | M_PER_MILE |
| ثابت نرخ تبدیل فوت به متر                                     | M_PER_FOOT |
| مایل که از ورودی خوانده می شود.                               | miles      |
| فوت که از ورودی خوانده می شود.                                | feet       |
| مقدار فوت و مایل که به متر تبدیل شده را نگه داری می کند.      | Meters     |
| مقادیر فوت و مایل که به کیلومتر تبدیل شده را نگه داری می کند. | kilometers |

```
M_PER_MILE = 1609.35;
M_PER_FOOT = 0.30480;
miles = int(input("Enter the number of miles: "))
feet = int(input("Enter the number of feet: "))
total_meters = miles * M_PER_MILE + feet * M_PER_FOOT
total_kilometers = total_meters / 1000;
kilometers = int(total_kilometers);
meters = (total_kilometers - kilometers) * 1000
print("The distance is ", kilometers, " kilometers, ", meters, " meters.")
```

دستورات اول و دوم، ثابت های M\_PER\_MILE و M\_PER\_FOOT را با مقادیر تعیین شده تعریف می کنند، دستورات سوم و چهارم، با پیغام های مناسب مقادیر را به مایل و فوت دریافت کرده، در متغیرهای miles و feet قرار می دهند، دستورات پنجم و ششم، به ترتیب کل مترها، کل کیلومترها را حساب می کنند و در total\_meters و total\_kilometers قرار می دهند، دستور هفتم، کل کیلو را به عدد صحیح تبدیل کرده، در kilometers قرار می دهد و دستور هشتم، تعداد مترهای باقی مانده را حساب می کند و در meters قرار می دهد و دستور آخر kilometers و meters را با پیغام های مناسب نمایش می دهد.

خروجی برنامه:

```
Enter the number of miles: 125
Enter the number of feet: 243
The distance is 201 kilometers, 242.8164000000948 meters.
```

**۴۸. برنامه‌ای که به متغیرهای محیطی دست‌یابی پیدا می‌کند.**

```
import os
# Access all environment variables
print('*-----*')
print("Environment is ", os.environ)
print('*-----*')
# Access a particular environment variable
print("home is ", os.environ['HOME'])
print('*-----*')
print("Path is ", os.environ['PATH'])
print('*-----*')
```

دستور اول، ماژول OS را به برنامه اضافه می‌کند، دستور دوم، یک توضیح است، دستور سوم، خط‌چین را چاپ می‌کند، دستور چهارم با خاصیت environ، محیط نصب پایتون را نمایش می‌دهد، دستور پنجم نیز خط‌چین را نمایش می‌دهد، دستور ششم، نیز یک توضیح است، دستور هفتم، با پیغام مناسب، خانه پایتون را نمایش می‌دهد، دستور هشتم، نیز یک خط‌چین را نمایش می‌دهد، دستور نهم، مسیر نصب پایتون را نمایش می‌دهد و بالاخره دستور آخر یک خط‌چین را نمایش می‌دهد.

**خروجی برنامه:**

```
*-----*
Environment is  environ({'COMSPEC': 'C:\\Windows\\system32\\cmd.exe',
'PROCESSOR_ARCHITECTURE': 'AMD64', 'LOGONSERVER': '\\\\FANSNO-PC', 'HOME':
'C:\\Users\\Fansno', 'LOCALAPPDATA': 'C:\\Users\\Fansno\\AppData\\Local',
'PROGRAMDATA': 'C:\\ProgramData', 'PROGRAMFILES': 'C:\\Program Files',
'OS': 'Windows_NT', 'PROGRAMW6432': 'C:\\Program Files',
'PROCESSOR_LEVEL': '20', 'COMMONPROGRAMFILES(X86)': 'C:\\Program Files
....
```

**۴۹. برنامه‌ای که نام کاربر فعلی را نمایش می‌دهد.**

```
import getpass
print("User name is ", getpass.getuser())
```

دستور اول، ماژول getpass را به برنامه اضافه می‌کند، دستور دوم، با فراخوانی getuser() و پیغام مناسب نام کاربر فعلی را نمایش خواهد داد.

**خروجی برنامه:**

```
User name is Fansno
```

**۵۰. برنامه‌ای که سه عدد را خوانده، بدون استفاده از دستورات شرطی و حلقه تکرار مرتب می‌کند.**

```
x = int(input("Input first number: "))
y = int(input("Input second number: "))
z = int(input("Input third number: "))
a1 = min(x, y, z)
a3 = max(x, y, z)
a2 = (x + y + z) - a1 - a3
print("Numbers in sorted order: ", a1, a2, a3)
```

دستورات اول تا سوم، سه عدد را خوانده، در متغیرهای x، y و z قرار می‌دهند، دستور چهارم، حداقل مقدار بین این سه عدد را در a<sub>1</sub> قرار می‌دهد، دستور پنجم، حداکثر مقدار بین این سه عدد را با تابع max() در a<sub>3</sub> قرار می‌دهد، دستور ششم، عدد وسطی (یعنی مجموع x+y+z اعداد) منهای a<sub>1</sub> (کوچک‌ترین عدد)، منهای a<sub>3</sub> (بزرگ‌ترین عدد) را در a<sub>2</sub> قرار می‌دهد و دستور آخر، با پیغام مناسب اعداد a<sub>1</sub>، a<sub>2</sub> و a<sub>3</sub> را نمایش می‌دهد.

**خروجی برنامه:**

```
Input second number: 8
```

```
Input third number: 9
Numbers in sorted order: 8 9 12
```

۵۱. برنامه‌ای که  $x$  و  $y$  را خوانده، حاصل عبارت مقابل را نمایش می‌دهد:

$$z = x^3 + 2x^2 + 3y - 5$$

```
x = int(input("Enter x:"))
y = int(input("Enter y:"))
z = x ** 3 + 2 * x ** 2 + 3 * y - 5
print("z is ", z)
```

دستورات اول و دوم، با پیغام‌های مناسب دو عدد را خوانده، در متغیرهای  $x$  و  $y$  قرار می‌دهند، دستور سوم، عبارت  $z$  را محاسبه می‌کند (عملگر  $**$ ، توان را محاسبه می‌نماید) و دستور آخر، با پیغام مناسب  $z$  را نمایش می‌دهد.

خروجی برنامه:

```
Enter x:12
Enter y:14
z is 2053
```

۵۲. برنامه‌ای که اطلاعات کپی‌رایت پایتون را نمایش می‌دهد.

```
import sys
print("\nPython Copyright Information")
print(sys.copyright)
```

دستور اول، ماژول `sys` را به برنامه اضافه می‌کند، دستور دوم، یک پیغام مناسب نمایش می‌دهد و دستور سوم، مقدار خاصیت `copyright` ماژول `sys` را نمایش می‌دهد.

خروجی برنامه:

```
Python Copyright Information
Copyright (c) ۲۰۰۱-۲۰۱۰ Python Software Foundation.
All Rights Reserved.
Copyright (c) ۲۰۰۰ BeOpen.com.
All Rights Reserved.
Copyright (c) ۱۹۹۵-۲۰۰۱ Corporation for National Research Initiatives.
All Rights Reserved.
Copyright (c) ۱۹۹۱-۱۹۹۵ Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.
```

۵۳. برنامه‌ای که دو عدد را خوانده، بدون استفاده از متغیر کمکی محتوی آن‌ها را تعویض می‌کند.

```
x = int(input("Enter x:"))
y = int(input("Enter y:"))
x = x + y
y = x - y
x = x - y
print("X = ", x, " Y = ", y)
```

دستورات اول و دوم، با پیغام‌های مناسب  $x$  و  $y$  را خوانده، دستور سوم،  $x+y$  را در  $x$  قرار می‌دهد (اگر  $x$  و  $y$  برابر ۱۴ و ۷۷ باشند، آنگاه  $x$  برابر با ۹۱ خواهد شد)، دستور چهارم،  $x-y$  (یعنی  $۹۱-۷۷$ ) را در  $y$  قرار می‌دهد، دستور پنجم  $x-y$  (یعنی  $۹۱-۱۴$ ) را در  $x$  قرار می‌دهد. همان‌طور که ملاحظه می‌کنید، جای  $x$  و  $y$  تعویض شده است. دستور آخر، مقادیر  $x$  و  $y$  را با پیغام‌های مناسب نمایش می‌دهد.

خروجی برنامه:

```
Enter x:14
Enter y:77
X = 77 Y = 14
```

۵۴. برنامه‌ای که تعداد آرگومان‌ها و آرگومان‌های خط فرمان را نمایش می‌دهد.

```
import sys
```



## آشنایی با پایتون ( دستورات ورودی و خروجی ) ۲۵

```
print("This is the name/path of the script:")
print("Number of arguments:",len(sys.argv))
print("Argument List:",str(sys.argv))
```

دستور اول، ماژول sys را به برنامه اضافه می کند، دستور دوم، یک پیام نمایش می دهد، دستور سوم، تعداد عناصر آرایه sys.argv را با متد len() و پیام مناسب نمایش می دهد و دستور سوم، با پیام مناسب مقدار آرایه sys.argv را به رشته تبدیل کرده، نمایش می دهد.

خروجی برنامه:

```
This is the name/path of the script:
Number of arguments: 1
Argument List: ['D:\\BookCSharp\\phyton-book\\program\\site\\py\\31.py']
```

۵۵. برنامه ای که قاعده و ارتفاع مثلث را خوانده، مساحت مثلث را محاسبه می کند (مساحت مثلث برابر با ارتفاع ضربدر نصف قاعده است).

```
p = int(input("Enter p:"))
h = int(input("Enter h:"))
s = p * h / 2
print("S is",s)
```

| متغیر | هدف    |
|-------|--------|
| h     | ارتفاع |
| p     | قاعده  |
| s     | مساحت  |

دستورات اول و دوم، به ترتیب قاعده و ارتفاع مثلث را خوانده، در p و h قرار می دهند، دستور دوم، مساحت مثلث را با فرمول قاعده \* نصف ارتفاع ( $p * h / 2$ ) حساب کرده، در s قرار می دهد و دستور چهارم، با پیام مناسب s را نمایش می دهد.

خروجی برنامه:

```
Enter p:12
Enter h:16
S is 96.0
```

۵۶. برنامه ای که نرخ حقوق به ازای هر ساعت و تعداد ساعت کارکرد یک کارمند را خوانده، میزان حقوق او را محاسبه می کند و نمایش می دهد (از حقوق ۱۰ درصد به عنوان مالیات کسر می شود. حقوق خالص برابر با حقوق منهای مالیات است).

| متغیر | هدف                             |
|-------|---------------------------------|
| h     | ساعت کارکرد کارمند              |
| r     | نرخ هر ساعت کارکرد              |
| tp    | مبلغ حقوق کارمند (مبلغ کل حقوق) |
| t     | مبلغ مالیات                     |
| pp    | حقوق خالص                       |

```
h = int(input("Enter hours:"))
r = int(input("Enter rate:"))
tp = h * r;
t = tp / 10;
pp = tp - t;
print("Total payment:", tp)
print("Tax:", t)
print("Payment:",pp )
```

## ۲۶ فصل اول

دستورات اول و دوم، با پیغام‌های مناسب ساعت کار کرد و نرخ هر ساعت را خوانده، در متغیرهای h و r قرار می‌دهند، دستور سوم، مبلغ حقوق را برابر ساعت کار کرد (h) ضرب در نرخ هر ساعت (r) قرار می‌دهد (tp، حقوق است)، دستور چهارم، مالیات (t) را برابر tp تقسیم بر ۱۰ قرار می‌دهد، دستور پنجم، حقوق خالص (pp) را برابر tp (حقوق) منهای مالیات (t) قرار می‌دهد و سه دستور آخر، با پیغام‌های مناسب حقوق، مالیات و حقوق خالص را نمایش می‌دهند.

خروجی برنامه:

```
Enter hours:10
Enter rate:5400
Total payment: 54000
Tax: 5400.0
Payment: 48600.0
```

۵۷. برنامه‌ای که مختصات دو نقطه را دریافت کرده، فاصله بین آن دو نقطه را محاسبه می‌نماید و نمایش می‌دهد.

$$\text{فاصله بین دو نقطه} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
x1 = int(input("Enter x1:"))
x2 = int(input("Enter x2:"))
y1 = int(input("Enter y1:"))
y2 = int(input("Enter y2:"))
d = ((x2-x1)**2+(y2-y1)**2)**0.5
print("Distance is ", d)
```

| متغیر          | هدف               |
|----------------|-------------------|
| X <sub>1</sub> | مختصات X نقطه اول |
| X <sub>2</sub> | مختصات X نقطه دوم |
| Y <sub>1</sub> | مختصات Y نقطه اول |
| Y <sub>2</sub> | مختصات Y نقطه دوم |

دستور اول تا چهارم، با پیغام‌های مناسب X<sub>1</sub>، X<sub>2</sub>، Y<sub>1</sub> و Y<sub>2</sub> را می‌خوانند، دستور پنجم، با فرمول داده‌شده، فاصله را حساب کرده، در d قرار می‌دهد (در اینجا \*\* عملگر توان است که اگر مقداری به توان ۰٫۵ برسد، جذر آن مقدار را حساب می‌کند) و دستور آخر، با پیغام مناسب d را نمایش می‌دهد.

خروجی برنامه:

```
Enter x1:10
Enter x2:20
Enter y1:30
Enter y2:40
Distance is 14.142135623730951
```

۵۸. زمینی به ابعاد ۱۶ × ۱۳ وجود دارد. موزاییک‌هایی به ابعاد ۳ × ۲، برای فرش کردن زمین نیاز است. اولاً تعیین

کنید چند موزاییک نیاز است. آیا قسمتی از زمین می‌ماند که با موزاییک فرش نشود؟

```
s1 = 13 * 16
print("s1 = ", s1)
s2 = 2 * 3
print("s2 = ", s2)
b = s1 // s2
a = s1 % s2;
print("b = ", b)
print("a = ", a)
```

| متغیر          | هدف                                  |
|----------------|--------------------------------------|
| S <sub>1</sub> | کل مساحت زمین                        |
| S <sub>2</sub> | مساحت یک موزاییک                     |
| b              | تعداد موزاییک‌های مورد نیاز          |
| a              | باقی‌مانده زمین که موزاییک نشده است. |

دستور اول ۱۶ × ۲ می‌کند تا کل ابعاد زمین را به دست آورده، در S<sub>1</sub> قرار می‌دهد دستور دوم، آن را با پیغام مناسب نمایش می‌دهد، دستور سوم، ۳ × ۲ را در S<sub>2</sub> قرار می‌دهد که مساحت یک موزاییک است، دستور چهارم، S<sub>2</sub> را با پیغام مناسب نمایش می‌دهد، دستور پنجم S<sub>1</sub> (کل مساحت زمین) تقسیم بر S<sub>2</sub> (مساحت هر

## ۲۷ آشنایی با پایتون ( دستورات ورودی و خروجی )

موزاییک) را در b قرار می دهد و سپس باقی مانده مساحتی که موزاییک نشد را از طریق  $s_2$  باقی مانده اش به  $s_1$  حساب کرده، در a قرار می دهد و در پایان، b و a را با پیغام مناسب نمایش می دهد.

خروجی برنامه:

```
s1 = 208
s2 = 6
b = 34
a = 4
```

۵۹. برنامه ای که نقطه تقاطع دو خط  $y = 3x + c$  و  $y = ax + b$  را نمایش می دهد (برنامه a، b و c را از ورودی می خواند). مختصات نقطه تقاطع به صورت زیر به دست می آید:

$$x = (b - c) / (3 - a)$$

$$y = 3 * x + c$$

```
a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = int(input("Enter c:"))
x = (b - c) / (3 - a)
y = 3 * x + c
print(", x, ", ", y, ")")
```

| متغیر | هدف                 |
|-------|---------------------|
| a     | ضریب X خط دوم       |
| b     | ثابت خط دوم         |
| c     | ثابت خط اول         |
| x     | مختصات X نقطه تقاطع |
| y     | مختصات Y نقطه تقاطع |

دستورات اول تا سوم، با پیغام های مناسب a، b و c را خوانده، دستورات چهارم و پنجم، x و y را محاسبه می کنند و دستور ششم، مقدار x و y را نمایش می دهد.

خروجی برنامه:

```
Enter a:10
Enter b:12
Enter c:14
( 14.857142857142858 , 0.2857142857142857 )
```

۶۰. برنامه ای که a و b را خوانده، مقدار  $y = a * \sin(b)$  را نمایش می دهد.

```
import math
a = int(input("Enter a:"))
b = int(input("Enter b:"))
y = a * math.sin((b * 3.14 / 180))
print("y is ", y)
```

دستور اول، ماژول math را به برنامه اضافه می کند، دستورات دوم و سوم، با پیغام های مناسب a و b را خوانده، دستور سوم، ابتدا b را به رادیان تبدیل کرده  $(b * 3.14 / 180)$  و سپس sin (سینوس) آن را حساب کرده، و در a ضرب می کند و نتیجه را در y قرار می دهد و در پایان، با پیغام مناسب y را نمایش می دهد.

خروجی برنامه:

```
Enter a:10
Enter b:15
y is 2.58690844053802
```

۶۱. برنامه ای که میزان حافظه مصرفی چند شیء را نمایش می دهد.

```
import sys
str1 = "one"
str2 = "four"
str3 = "three"
print("Memory size of "+str1+" = "+str(sys.getsizeof(str1))+ " bytes")
print("Memory size of "+str2+" = "+str(sys.getsizeof(str2))+ " bytes")
```

## ۲۸ فصل اول

```
print("Memory size of '"+str3+"' = "+str(sys.getsizeof(str3))+ " bytes")
```

دستور اول، ماژول sys را برنامه اضافه می کند، دستورات دوم تا چهارم، سه متغیر با مقادیر one، four و three را تعریف می کنند و دستورات بعدی با پیام های مناسب اندازه این متغیرها را با متد getsizeof () نمایش می دهند.

**خروجی برنامه:**

```
Memory size of 'one' = 28 bytes
Memory size of 'four' = 29 bytes
Memory size of 'three' = 30 bytes
```

**۶۲. برنامه ای که نام میزبانی را که روال آن در حال اجرا است، نمایش می دهد.**

```
import socket
host_name = socket.gethostname()
print("Host name:", host_name)
```

دستور اول، ماژول socket را به برنامه اضافه می کند، دستور دوم، با متد gethostname() نام میزبان برنامه فعلی در حال اجرا را برگردانده، در متغیر host\_name قرار می دهد و دستور آخر، مقدار host\_name را با پیام مناسب نمایش می دهد.

**خروجی برنامه:**

```
Host name: Fansno-PC
```

**۶۳. برنامه ای که نام فایل را از مسیر جدا کرده، نمایش می دهد.**

```
import os
print(os.path.basename('/users/system1/student1/homework-1.py'))
```

دستور اول، ماژول os را به برنامه اضافه می کند، دستور دوم، با متد basename() نام فایل را جدا کرده، با پیام مناسب نمایش می دهد.

**خروجی برنامه:**

```
homework-1.py
```

**۶۴. برنامه ای که یک کاراکتر را خوانده، کد اسکی آن را نمایش می دهد (در این برنامه متغیر ch کاراکتر خوانده شده را نگهداری می کند).**

```
ch = input("Enter a char:")
print(ch, " ", ord(ch))
```

دستور اول، کاراکتری را خوانده، در ch قرار می دهد، دستور دوم، کاراکتر خوانده شده و کد اسکی آن (ord(ch)) را نمایش می دهد.

**خروجی برنامه:**

```
Enter a char:A
A      65
```

**۶۵. برنامه ای که اطلاعات محیط کاربر را نمایش می دهد.**

```
import os
print(os.environ)
```

دستور اول، ماژول os را به برنامه اضافه می کند، دستور دوم، با خاصیت environ از os، محیط کاربر را نمایش می دهد.

**خروجی برنامه:**

```
environ({'PUBLIC': 'C:\\Users\\Public', 'JAVA_HOME': 'C:\\Program
Files\\Java\\jdk1.8.0_102\\jre', 'PROCESSOR_ARCHITECTURE': 'AMD64',
'WINDOWS_TRACING_LOGFILE':
'C:\\BVTBin\\Tests\\installpackage\\csilogfile.log', 'LOGONSERVER':
```

```
'\\\\FANSNO-PC', 'PROCESSOR_IDENTIFIER': 'AMD64 Family 20 Model 2 Stepping
0, AuthenticAMD', 'USERNAME': 'Fansno', 'WINDIR': 'C:\\Windows',
'HOMEPath': '\\Users\\Fansno', 'ALLUSERSPROFILE': 'C:\\ProgramData',
'PROCESSOR_LEVEL': '20', 'HOME': 'C:\\Users\\Fansno', 'VS120COMNTOOLS':
'C:\\Program Files (x86)\\Microsoft Visual Studio 12.0\\Common7\\Tools\\',
'COMSPEC': 'C:\\Windows\\system32\\cmd.exe', ...
```

۶۶. برنامه‌ای که ابتدا عددی را خوانده، سپس  $n$  را می‌خواند و مقدار  $n$  امین بیت عدد اول خوانده‌شده را نمایش می‌دهد.

```
num = int(input("Enter a number: "))
n = int(input("Enter nth bit to check (0-31): "))
bitStatus = (num >> n) & 1
print("The", n, " bit is set to ", bitStatus)
```

در این برنامه  $num$  (عدد ورودی)،  $n$  (بیت موردنظر) و  $bitState$  (وضعیت یا مقدار  $n$  امین بیت

$num$ ) را داریم.

این برنامه برای خواندن بیت  $n$ ام یک عدد، ابتدا عدد موردنظر را  $n$  بیت با سمت راست شیفت داده و

با ۱ و بیتی می‌نماید.

خروجی برنامه:

```
Enter a number: 121
Enter nth bit to check (0-31): 4
The 4 bit is set to 1
```

۶۷. برنامه‌ای که ابتدا عددی را خوانده، سپس عدد دیگر  $n$  را می‌خواند و  $n$  امین بیت عدد را برابر ۱ (یک) قرار می‌دهد.

```
num = int(input("Enter a number: "))
n = int(input("Enter nth bit to set (0-31): "))
newNum = (1 << n) | num
print("Bit set successfully.")
print("Number before setting ", n, " bit:", num, "(in decimal)")
print("Number after setting ", n, " bit:", newNum, "(in decimal)")
```

در این برنامه متغیرهای  $num$  (عدد ورودی)،  $n$  (بیتی که باید یک شود) و  $newNum$  (عدد جدید

پس از یک شدن بیت  $n$ ام) را داریم. این برنامه برای یک کردن بیت  $n$ ام عدد  $num$ ، یک را  $n$  بیت به

سمت چپ شیفت داده با  $num$  یا بیتی می‌کند.

خروجی برنامه:

```
Enter a number: 10
Enter nth bit to set (0-31): 6
Bit set successfully.
Number before setting 6 bit: 10 (in decimal)
Number after setting 6 bit: 74 (in decimal)
```

۶۸. برنامه‌ای که ابتدا عددی را خوانده، سپس  $n$  را می‌خواند و  $n$  امین بیت عدد خوانده‌شده را صفر می‌نماید. به‌عنوان مثال، شکل زیر امین بیت عدد ۵۵ را صفر می‌نماید.

```

55 ← num (in decimal)
-----
4 ← n (in decimal)
0000 0001 ← 1 (in binary)
-----
0001 0000 ← 1 << n
1110 1111 ← ~(1 << n)
0011 0111 ← num (in binary)
-----
0010 0111 ← num & (~ (1 << n))

```

```

num = int(input("Enter a number: "))
n = int(input("Enter nth bit to set (0-31): "))
newNum = num & (~ (1 << n))
print("Bit cleared successfully.")
print("Number before clearing ", n, " bit:", num, "(in decimal)")
print("Number after clearing ", n, " bit:", newNum, "(in decimal)")

```

در این برنامه `num` (عدد ورودی)، `n` (عددی که شماره بیت `num` آن باید صفر شود) و `newNum` (عدد جدید) را داریم.

خروجی برنامه:

```

Enter a number: 1234
Enter nth bit to set (0-31): 6
Bit cleared successfully.
Number before clearing 6 bit: 1234 (in decimal)
Number after clearing 6 bit: 1170 (in decimal)

```

۶۹. برنامه‌ای که ابتدا عددی را خوانده، سپس عدد `n` را می‌خواند و `n` امین بیت عدد خوانده شده را معکوس می‌کند. یعنی، چنانچه این بیت یک باشد به صفر و اگر صفر باشد به یک تبدیل می‌نماید.

```

num = int(input("Enter a number: "))
n = int(input("Enter nth bit to set (0-31): "))
newNum = num ^ (1 << n)
print("Bit toggled successfully.")
print("Number before toggling ", n, " bit:", num, "(in decimal)")
print("Number after toggling ", n, " bit:", newNum, "(in decimal)")

```

در این برنامه `num` (عدد ورودی)، `n` (شماره بیتی از عدد `num` است که باید معکوس گردد) و `newNum` (عدد جدید) را داریم.

این برنامه ابتدا عدد یک (`1`) را `n` بیت به سمت چپ شیفت می‌دهد و با `num` یا بیتی می‌نماید.

خروجی برنامه:

```

Enter a number: 123
Enter nth bit to set (0-31): 6
Bit toggled successfully.
Number before toggling 6 bit: 123 (in decimal)
Number after toggling 6 bit: 59 (in decimal)

```

۷۰. برنامه‌ای که دو عدد را خوانده، با استفاده از عملگر یا انحصاری بیتی آن‌ها را باهم تعویض می‌نماید و نمایش می‌دهد. مراحل انجام این کار در شکل زیر آمده است:

### آشنایی با پایتون ( دستورات ورودی و خروجی ) ۳۱

$$\begin{array}{r}
 a \quad 0001\ 1010 \\
 b \wedge 0100\ 0001 \\
 \hline
 x \quad 0101\ 1011
 \end{array}
 \quad
 \begin{array}{r}
 b \quad 0100\ 0001 \\
 x \wedge 0101\ 1011 \\
 \hline
 a \quad 0001\ 1010
 \end{array}
 \quad
 \begin{array}{r}
 a \quad 0001\ 1010 \\
 x \wedge 0101\ 1011 \\
 \hline
 b \quad 0100\ 0001
 \end{array}$$

```

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
print("Num1 = ", num1)
print("Num2 = ", num2)
num1 ^= num2
num2 ^= num1
num1 ^= num2
print("Num1 after swapping = ", num1)
print("Num2 after swapping = ", num2)

```

در این برنامه num<sub>۱</sub> (عدد اول) و num<sub>۲</sub> (عدد دوم) هستند.

خروجی برنامه:

```

Enter first number: 10
Enter second number: 20
Num1 = 10
Num2 = 20
Num1 after swapping = 20
Num2 after swapping = 10

```

۷۱. برنامه‌ای که سه عدد را به عنوان قاعده کوچک، قاعده بزرگ و ارتفاع یک دوزنقه خوانده، مساحت دوزنقه را نمایش می‌دهد. مساحت دوزنقه برابر با:

$$\text{مساحت دوزنقه} = \frac{(\text{قاعده بزرگ} + \text{قاعده کوچک}) * \text{ارتفاع}}{2}$$

در این برنامه a، b، c به ترتیب قاعده کوچک، قاعده بزرگ و ارتفاع دوزنقه هستند.

```

a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = int(input("Enter c:"))
print("Area is ", 1 / 2 * (a + b) * c)

```

خروجی برنامه:

```

Enter a:12
Enter b:14
Enter c:16
Area is 208.0

```

۷۲. برنامه‌ای که عددی را خوانده، تمام بیت‌های عدد خوانده شده را معکوس می‌کند. یعنی، چنانچه بیت یک باشد به صفر و اگر صفر باشد، به یک تبدیل می‌نماید.

```

num = int(input("Enter a number: "))
flippedNumber = ~num
print("Original number = ", num, "(in decimal)")
print("Number after bits are flipped = ", flippedNumber, " (in decimal)")

```

در این برنامه num (عدد ورودی) و flippedNumber (عددی است که باید بیت‌های آن

معکوس گردد) را داریم.

خروجی برنامه:

```

Enter a number: 120
Original number = 120 (in decimal)
Number after bits are flipped = -121 (in decimal)

```

**۷۳. برنامه‌ای که کم‌ترین، بیش‌ترین مقدار اعداد صحیح، اعداد صحیح با طول بلند و اعداد اعشاری را نمایش می‌دهد.**

```
import sys
print("Float value information: ",sys.float_info)
print("\nInteger value information: ",sys.int_info)
print("\nMaximum size of an integer: ",sys.maxsize)
```

دستور اول، ماژول sys را به برنامه اضافه می‌کند، دستور دوم، با خاصیت float\_info از sys، اطلاعات مربوط به نوع float را نمایش می‌دهد، دستور سوم، با خاصیت int\_info مربوط به sys اطلاعات نوع int را نمایش می‌دهد و دستور آخر، با خاصیت maxsize مربوط به sys بیش‌ترین مقدار عدد صحیح را نمایش می‌دهد.

**خروجی برنامه:**

```
Float value information: sys.float_info(max=1.7976931348623157e+308,
max_exp=1024, max_10_exp=308, min=2.2250738585072014e-308, min_exp=-1021,
min_10_exp=-307, dig=15, mant_dig=53, epsilon=2.220446049250313e-16,
radix=2, rounds=1)
Integer value information: sys.int_info(bits_per_digit=30, sizeof_digit=4)
Maximum size of an integer: 9223372036854775807
```

**۷۴. برنامه‌ای که زمان فعلی سیستم را نمایش می‌دهد.**

```
import datetime
print("Current time is ", datetime.datetime.now().time())
```

دستور اول، ماژول datetime را به برنامه اضافه می‌کند، دستور دوم، از طریق متد now() و time() زمان فعلی سیستم را نمایش می‌دهد now() زمان و تاریخ را برمی‌گرداند و time() از آن زمان را جدا می‌نماید.

**خروجی برنامه:**

```
Current time is 19:54:58.478599
```

**۷۵. برنامه‌ای که n ثانیه به زمان فعلی اضافه می‌کند.**

```
import datetime
n = int(input("Enter n:"))
x = datetime.datetime.now()
y = x + datetime.timedelta(0, n)
print("Current time is ", x.time())
print("New Time is", y.time())
```

دستور اول، ماژول datetime را به برنامه اضافه می‌کند، دستور دوم، n را می‌خواند، دستور سوم، از طریق متد now()، زمان و تاریخ فعلی را در x قرار می‌دهد، دستور سوم، از طریق متد timedelta()، n ثانیه به زمان و تاریخ x اضافه کرده، در y قرار می‌دهد. دستور چهارم، بخش زمان x را از طریق متد time() جدا کرده و نمایش می‌دهد و دستور آخر، بخش زمان y را جدا کرده و نمایش خواهد داد.

**خروجی برنامه:**

```
Enter n:3600
Current time is 20:03:49.051124
New Time is 21:03:49.051124
```

**۷۶. برنامه‌ای که زاویه‌های را به درجه دریافت کرده، به رادیان تبدیل می‌نماید و نمایش می‌دهد.**

```
pi=22/7
degree = float(input("Input degrees: "))
radian = degree*(pi/180)
print("Radian is ", radian)
```



### آشنایی با پایتون ( دستورات ورودی و خروجی ) ۳۳

دستور اول، pi را برابر ۲۲ تقسیم ۷ قرار می دهد، دستور دوم، زاویه را به درجه خوانده، در degree قرار می دهد، دستور سوم، pi \* degree تقسیم بر ۱۸۰ کرده تا به رادیان تبدیل شود، سپس آن را در متغیر radian قرار می دهد و دستور آخر با پیغام مناسب radian را نمایش می دهد.

خروجی برنامه:

```
Input degrees: 110
Radian is 1.9206349206349207
```

۷۷. برنامه ای که زاویه های را به رادیان دریافت کرده، به درجه تبدیل می نماید و نمایش می دهد.

```
pi=22/7
radian = float(input("Input radians: "))
degree = radian*(180/pi)
print("Degree is ", degree)
```

دستور اول pi را برابر ۲۲ تقسیم بر ۷ قرار می دهد، دستور سوم، با پیغام مناسب زاویه را به رادیان می خواند، دستور سوم، با فرمول radian \* ۱۸۰ تقسیم بر pi، زاویه رادیان را به درجه تبدیل کرده و در متغیر degree قرار می دهد، دستور آخر با پیغام مناسب degree را نمایش می دهد.

خروجی برنامه:

```
Input radians: 1.8
Degree is 103.0909090909091
```

۷۸. برنامه ای که سه مقدار را خوانده، نوع آن ها را نمایش می دهد.

```
a = int(input("Enter a: "))
b = float(input("Enter b: "))
c = input("Enter c: ")
print("Type a is " , type(a))
print("Type b is " , type(b))
print("Type c is " , type(c))
```

| متغیر | هدف                           |
|-------|-------------------------------|
| a     | متغیر ورودی از نوع عدد صحیح   |
| b     | متغیر ورودی از نوع عدد اعشاری |
| c     | متغیر ورودی از نوع رشته ای    |

دستورات اول تا سوم، به ترتیب سه مقدار عددی صحیح، عددی اعشاری و رشته ای را خوانده، در متغیرهای a، b و c قرار می دهند و سه دستور بعدی با تابع type()، نوع این متغیرها را با پیغام های مناسب نمایش می دهند.

خروجی برنامه:

```
Enter a: 10
Enter b: 12.5
Enter c: string
Type a is <class 'int'>
Type b is <class 'float'>
Type c is <class 'str'>
```

## ساختار تصمیم و حلقه تکرار

۱. برنامه‌ای که شماره دانشجویی و معدل تعداد N دانشجو را از ورودی خوانده، دانشجویی که دومین معدل را از نظر بزرگی دارد، پیدا می‌کند و به خروجی می‌برد. (توضیح: چون هیچ‌یک از دانشجویان معدل‌شان کوچک‌تر از صفر نمی‌باشد، لذا اولین بار معدل دو دانشجویی که بالاترین معدل را دارند، صفر در نظر گرفتیم.)

```

id1 = -1
max1 = -1
max2 = -1
id2 = -1
n = int(input("Enter n:"))
if n < 2:
    print("Please enter a number greater then 1")
else:
    for i in range(1, n+1):
        id = int(input("Enter id:"))
        aver = float(input("Enter average:"))
        if aver > max1:
            id2=id1
            max2 = max1
            max1 = aver
            id1 = id
        else:
            if aver>max2:
                max2=aver
                id2=id
    print ("Max2 = ", max2, '\t', "Id2 =", id2)

```

| متغیر            | هدف                                   |
|------------------|---------------------------------------|
| n                | تعداد دانشجویان                       |
| max <sub>۱</sub> | اولین معدل                            |
| max <sub>۲</sub> | دومین معدل                            |
| i                | شمارنده                               |
| id <sub>۱</sub>  | شماره دانشجویی که اولین معدل را دارد. |
| id <sub>۲</sub>  | شماره دانشجویی که دومین معدل را دارد. |
| aver             | معدل دانشجو                           |
| Id               | شماره دانشجو                          |

دستورات اول تا چهارم، متغیرهای id<sub>۱</sub>، max<sub>۱</sub>، max<sub>۲</sub> و id<sub>۲</sub> را مقدار ۱- می‌دهند (چون هیچ دانشجویی معدل آن‌ها کم‌تر از ۰ نیست، پس همه معدل‌ها بزرگ‌تر از ۱- هستند و هیچ شماره دانشجویی ۱- نمی‌باشد). دستور پنجم، با پیغام مناسبی n را خوانده، دستور ششم، اگر n کوچک‌تر از ۱ باشد، پیغام مناسب نمایش می‌دهد (چون حداقل دو دانشجو باید اطلاعاتشان وارد گردد)، در غیر این صورت (اگر n بزرگ‌تر یا مساوی ۲ باشد)، اعمال زیر را انجام می‌دهد:

۱. یک حلقه از ۱ تا n با شمارنده i ایجاد می‌کند.
۲. در این حلقه با یک پیغام مناسب یک شماره دانشجویی را خوانده، در id قرار می‌دهد.
۳. با یک پیغام مناسب، یک معدل را خوانده، در متغیر aver قرار می‌دهد.

### ساختار تصمیم و حلقه تکرار ۳۵

۴. اگر  $aver$  بزرگ‌تر از  $max_2$  باشد (معدل خوانده شده بزرگ‌تر از بزرگ‌ترین معدل باشد)، آنگاه ابتدا،  $id_1$  را در  $id_2$  قرار می‌دهد، سپس،  $max_1$  را در  $max_2$  قرار می‌دهد (بزرگ‌ترین معدل قبلی و شماره دانشجویی آن، الآن دومین معدل و دومین شماره دانشجویی خواهد شد)، در ادامه،  $aver$  را در  $max_1$  قرار می‌دهد و  $id_1$  را در  $id_2$  قرار خواهد داد.
۵. وگرنه، اگر  $aver$  (معدل خوانده شده فعلی) بزرگ‌تر از  $max_2$  (دومین معدل) باشد، آنگاه ابتدا،  $aver$  را در  $max_2$  قرار می‌دهد و سپس  $id_1$  را در  $id_2$  قرار خواهد داد.
- در پایان حلقه،  $max_2$  و  $id_2$  را با پیغام مناسب نمایش می‌دهد.
- خروجی برنامه:**

```
Enter n:3
Enter id:1
Enter average:17.5
Enter id:2
Enter average:12
Enter id:3
Enter average:15
Max2 = 15.0 Id2 = 3
```

۲. برنامه‌ای که اعدادی را از ورودی خوانده تشخیص می‌دهد که آیا اعداد موردنظر کامل هستند یا خیر. عددی کامل است که مجموع مقسوم‌علیه‌های آن (به‌جز خودش) برابر با آن عدد باشد، پس از بررسی هر عدد از کاربر سؤال کند که می‌خواهد به کارش ادامه دهد یا خیر.

```
while True:
    num = int(input("Enter a number:"))
    sum = 0
    for i in range(1, num):
        if (num % i == 0):
            sum += i
    if (sum == num):
        print('\t', "Perfect")
    else:
        print('\t', "Not perfect")
    yes=input("Continue ?")
    if (yes[0] == 'N' or yes[0] == 'n'):
        break
```

| متغیر | هدف                 |
|-------|---------------------|
| num   | اعداد موردنظر       |
| i     | شمارنده از ۱ تا num |
| sum   | مجموع مقسوم‌علیه‌ها |
| yes   | تعیین ادامه کار     |

- دستور اول، یک حلقه بی‌نهایت ایجاد می‌کند که در این حلقه اعمال زیر انجام می‌شود:
- با پیغام مناسب یک عدد را خوانده، در  $num$  قرار می‌دهد.
  - مجموع مقسوم‌علیه‌های عدد (یعنی  $sum$ ) را برابر صفر قرار می‌دهد.
  - یک حلقه ایجاد می‌کند که شمارنده آن از ۱ تا  $num-1$  شمارش می‌کند، در این حلقه، اگر  $num$  بر  $i$  بخش پذیر باشد ( $num \% i == 0$ )، آنگاه،  $i$  مقسوم‌علیه  $num$  است، پس  $sum$  را با  $i$  جمع کرده، در  $sum$  قرار می‌دهد.
  - اگر  $sum$  برابر  $num$  باشد، آنگاه عبارت "Perfect" را نمایش خواهد داد.
  - با یک پیغام مناسب، یک رشته را خوانده، در  $yes$  قرار می‌دهد.
  - اگر کاراکتر اول  $yes$  (یعنی  $yes[0]$ ) برابر 'N' یا 'n' باشد، با دستور  $break$  حلقه  $while$  را خاتمه می‌دهد (یعنی، کاربر نمی‌خواهد ادامه بدهد).

خروجی برنامه:

```
Enter a number:6
    Perfected
Continue ?y
Enter a number:12
    Not perfect
Continue ?n
```

۳. برنامه‌ای که تعداد  $n$  جمله از سری فیبوناچی را تولید می‌کند.

سری فیبوناچی = ۱ ۱ ۲ ۳ ۵ ۸ ۱۳ ...

```
f1 = 1
f2 = 1
n = int(input("Enter a number:"))
if n == 1:
    print(f1)
    exit(0)
elif n == 2:
    print(f1)
    print(f2)
else:
    print(f1)
    print(f2)
    i = 3
    while i <= n:
        f3 = f1 + f2
        print(f3, " ")
        f1 = f2
        f2 = f3
        i = i + 1
```

| متغیر          | هدف         |
|----------------|-------------|
| i              | شمارنده     |
| f <sub>۱</sub> | جملات سری   |
| f <sub>۲</sub> | جملات سری   |
| f <sub>۳</sub> | جملات سری   |
| n              | تعداد جملات |

دستورات اول و دوم، متغیرهای  $f_1$  (اولین عدد فیبوناچی در گام اول و در گام‌های بعدی دو عدد قبلی فیبوناچی)،  $f_2$  (دومین عدد فیبوناچی در گام اول و در گام‌های بعدی دو قبلی فیبوناچی) را برابر ۱ قرار می‌دهند، دستور سوم، با پیغام مناسب  $n$  را می‌خواند، دستور چهارم، اگر  $n=1$  باشد،  $f_1$  را نمایش می‌دهد، وگرنه، اعمال زیر را انجام می‌دهد:

۱.  $f_1$  و  $f_2$  را نمایش می‌دهد.

۲.  $i$  را بر ۳ قرار می‌دهد (چون، عدد بعدی، سومین عدد فیبوناچی است که می‌خواهد نمایش دهد).

۳. تا زمانی که  $i$  کوچک‌تر یا مساوی  $n$  باشد، ابتدا،  $f_1$  به علاوه  $f_2$  را در  $f_3$  قرار می‌دهد تا عدد فیبوناچی بعدی به دست آید، سپس  $f_3$  را نمایش می‌دهد، در ادامه، ابتدا،  $f_2$  را در  $f_1$  قرار می‌دهد (عدد فیبوناچی قبلی را در دو قبلی فیبوناچی برای مرحله بعد قرار می‌دهد، در پایان،  $i$  را یک واحد اضافه می‌کند تا اعداد فیبوناچی تولیدشده یکی اضافه گردد.

خروجی برنامه:

```
Enter a number:6
1
1
2
3
5
8
```

۴. برنامه‌ای که رشته‌ای را خوانده، کاراکترهای رشته را با یک فاصله از یکدیگر نمایش می‌دهد (هدف برنامه پیمایش رشته با for می‌باشد).

```
s = input("Enter a string:")
for i in s:
    print(i, end = ' ')
```

| متغیر | هدف             |
|-------|-----------------|
| s     | رشته خوانده شده |
| i     | هر کاراکتر رشته |

دستور اول، با پیغام مناسب یک رشته را خوانده، در s قرار می‌دهد، دستور دوم، با استفاده از یک حلقه for کاراکترهای رشته را پیمایش کرده، در i قرار می‌دهد و با دستور print() مقدار i را نمایش می‌دهد (عبارت 'end=' می‌شود کنترل چاپ به سطر بعدی نرود و یک فاصله بین چاپ کاراکترها قرار گیرد).

خروجی برنامه:

```
Enter a string:Fanavarienovin
F a n a v a r i e n o v i n
```

۵. فرض کنید بخواهید از بانک ۱۰۰۰۰۰۰۰ ریال وام با بهره ۱۵٪ با مدت بازپرداخت ۱۸ ماهه دریافت کنید، بهره وام به صورت زیر محاسبه می‌شود:

$$\text{بهره وام} = \frac{10000000 \times 19 \times 15}{2400} = \frac{10000000 \times 19 \times 15}{2400} = 1187500$$

مبلغ بهره (۱۱۸۷۵۰۰) ریال است. حال چنانچه متقاضی ۱۰۰۰۰۰۰۰ ریال نیاز داشته باشد، چقدر وام باید به او پرداخت شود. برنامه‌ای که مبلغ مورد نیاز متقاضی (اصل وام)، تعداد اقساط و مبلغ بهره را دریافت می‌کند، سپس وامی که باید به متقاضی پرداخت شود و قسط هر ماه را چاپ می‌کند. (توضیح: مبلغ وام پرداختی به صورت زیر محاسبه می‌گردد).

$$\text{وام اصل} + \text{بهره وام} = \text{مبلغ وام پرداختی}$$

تذکر: برنامه تا هر زمان که کاربر بخواهد ادامه می‌یابد.

```
while True :
    m = int(input("Enter m:"))
    tedad = int(input("Enter tedad:"))
    s = int (input("Enter s:"))
    k = m + (m*(tedad+1)*s) /2400
    p = k / tedad
    print("k = ", k, '\t', p)
    ansi=input("Do you want to continue(y/n):")
    if ansi[0] == 'n':
        break
```

| متغیر | هدف            |
|-------|----------------|
| m     | اصل وام        |
| s     | نرخ بهره       |
| k     | کل وام پرداختی |
| tedad | تعداد اقساط    |
| p     | پرداختی هر قسط |
| ansi  | شرط ادامه      |

دستور اول، یک while (حلقه) بی‌نهایت ایجاد می‌کند که در این حلقه اعمال زیر انجام می‌شود:

- با پیغام مناسب مبلغ مورد نیاز را خوانده، در m قرار می‌دهد.
- با پیغام مناسب دیگر، تعداد اقساط را خوانده، در tedad قرار می‌دهد.
- با پیغام مناسب دیگر، نرخ بهره را خوانده، در s قرار می‌دهد.
- طبق فرمول مبلغ کل وام را محاسبه کرده، در k قرار می‌دهد.
- مبلغ کل پرداختی (k) را تقسیم بر تعداد اقساط (tedad) کرده تا مبلغ هر قسط (یعنی p) به دست آید.
- k و p را با پیغام مناسب نمایش می‌دهد.

۷. با یک پیغام مناسب از کاربر می‌خواهد که آیا ادامه می‌دهد یا نه؟ رشته دریافت شده را در ansi قرار می‌دهد.

۸. اگر اولین کاراکتر رشته ansi برابر 'n' باشد، با دستور break حلقه را خاتمه می‌دهد.

خروجی برنامه:

```
Enter m:5000000
Enter tedad:60
Enter s:18
k = 72875000.0      1214583.3333333333
Do you want to continue(y/n):y
Enter m:10000000
Enter tedad:36
Enter s:18
k = 12775000.0      354861.1111111111
Do you want to continue(y/n):y
Enter m:10000000
Enter tedad:48
Enter s:24
k = 14900000.0      310416.6666666667
Do you want to continue(y/n):n
```

۶. برنامه‌ای که سال تولد کاربر و سال فعلی را از ورودی خوانده، مشخص می‌کند چند سال، چند ماه، چند روز، چند ساعت، چند دقیقه و چند ثانیه عمر کرده است.

```
byy =int(input("Enter birth date(year):"))
bmm =int(input("Enter birth date(month):"))
bdd =int(input("Enter birth date(day):"))
cyy =int(input("Enter current date(year):"))
cmm =int(input("Enter current date(month):"))
cdd =int(input("Enter current date(day):"))
if cdd < bdd:
    cmm -= 1
    cdd += 30
day = cdd - bdd
if cmm < bmm:
    cyy -= 1
    cmm += 12
month = cmm - bmm
year = cyy - byy
days = day + month * 30 + year * 365
hh = days * 24
mm = hh * 60
ss = mm * 60
print("Old is: ",year,"/",month,"/", day)
print("Hour is(hh:mm:ss): ",hh,":",mm,":",ss)
```

| متغیر | هدف                            |
|-------|--------------------------------|
| byy   | سال تولد                       |
| bmm   | ماه تولد                       |
| bdd   | روز تولد                       |
| cyy   | سال جاری                       |
| cmm   | ماه جاری                       |
| cdd   | روز جاری                       |
| year  | سال‌هایی که زندگی کرده         |
| month | ماه‌هایی که زندگی کرده         |
| day   | روزهایی که زندگی کرده          |
| hh    | تعداد ساعتی که زندگی کرده      |
| mm    | تعداد دقیقاتی که زندگی کرده    |
| ss    | تعداد ثانیه‌هایی که زندگی کرده |
| days  | تعداد کل روزهایی که زندگی کرده |

دستورات اول تا سوم، به ترتیب با پیغام‌های مناسب، سال، ماه، روز تولد را خوانده، در متغیرهای byy، bmm و bdd قرار می‌دهند. دستورات چهارم تا ششم، با پیغام‌های مناسب به ترتیب سال، ماه و روز فعلی را خوانده، در متغیرهای cyy، cmm و cdd قرار می‌دهند. دستور هفتم، اگر روز فعلی (cdd) کوچک‌تر از روز تولد (bdd) باشد، ابتدا از ماه فعلی cmm یک واحد کم می‌کند و به cdd، به اندازه ۳۰ روز اضافه می‌کند (چون هر ماه ۳۰ روز است)، دستور هشتم، bdd را از cdd کم کرده، در day قرار می‌دهد، دستور نهم، اگر cmm

### ساختار تصمیم و حلقه تکرار ۳۹

(ماه فعلی) کوچک‌تر از ماه تولد (bmm) باشد، ابتدا از cyy یک واحد کم می‌کند و به cmm، ۱۲ واحد اضافه می‌کند (یک سال ۱۲ ماه است). دستور دهیم، bmm را از cmm کم کرده تا تعداد ماه‌های زندگی کرده به دست آید و آن را در month قرار می‌دهد، دستور یازدهم، ابتدا، day را با month ضرب در ۳۰ جمع کرده، سپس نتیجه آن را با year ضرب در ۳۶۵ ضرب کرده تا تعداد کل روزهایی که زندگی کرده است را به دست آورد و در year قرار می‌دهد، دستور دوازدهم، days را ضرب در ۲۴ کرده تا تعداد ساعاتی که زندگی کرده است را به دست آورد و در hh قرار می‌دهد، دستور سیزدهم، hh ضرب در ۶۰ (هر ساعت ۶۰ دقیقه است) کرده تا تعداد دقیقگی که زندگی نموده است را به دست آورد و در mm قرار می‌دهد، دستور چهاردهم، mm را ضرب در ۶۰ می‌کند تا تعداد ثانیه‌هایی که زندگی کرده است را به دست آورد و در ss قرار می‌دهد، دستور پانزدهم، تعداد سال (year)، تعداد ماه (month) و تعداد روز (day) را با پیغام مناسب نمایش می‌دهد و دستور آخر، تعداد ساعات، دقیقگی و ثانیه‌ها را با پیغام مناسب نمایش خواهد داد.

**خروجی برنامه:**

```
Enter birth date(year):1348
Enter birth date(month):10
Enter birth date(day):4
Enter current date(year):1398
Enter current date(month):6
Enter current date(day):6
Old is: 49 / 8 / 2
Hour is(hh:mm:ss): 435048 : 26102880 : 1566172800
```

**۷. برنامه‌ای که سه مقدار را از ورودی خوانده، آن‌ها را به ترتیب صعودی در خروجی چاپ می‌کند.**

```
a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = int(input("Enter c:"))
if a > b:
    temp = a
    a = b
    b = temp
if a > c:
    temp = a
    a = c
    c = temp
if a > c:
    temp = a
    a = c
    c = temp
if b > c:
    temp = b
    b = c
    c = temp
print("Sorted is ", a, b, c)
```

| متغیر   | هدف              |
|---------|------------------|
| a, b, c | اعداد خوانده شده |
| temp    | متغیر کمکی       |

دستورات اول تا سوم، با پیغام‌های مناسب سه عدد را خوانده، در a، b و c قرار می‌دهند، دستور چهارم، اگر a بزرگ‌تر از b باشد، مقادیر a و b را عوض می‌کند، دستور پنجم، اگر a بزرگ‌تر از c باشد، مقادیر a و c را تعویض می‌کند، دستور ششم، اگر b بزرگ‌تر از c باشد، مقادیر a و c را عوض می‌کند و دستور هفتم، با پیغام مناسب، مقادیر a، b و c را نمایش می‌دهد.

**خروجی برنامه:**

```
Enter a:12
Enter b:8
Enter c:9
```

Sorted is 8 9 12

۸. برنامه‌ای که شماره کارمندی، ساعت کارکرد و دستمزد ساعتی کارکنان موسسه‌ای را خوانده، حقوق آن‌ها را محاسبه می‌کند. اگر کارمندی بیش از ۴۰ ساعت کار کرده باشد، اضافه‌کار به او تعلق می‌گیرد. به ازای هر ساعت کاری، ۵٫۰ دستمزد ساعتی (نصف دستمزد ساعتی) به‌عنوان اضافه‌کاری پرداخت می‌شود.

```
n = int(input("Enter n:"))
i = 1
while i <= n:
    id = int(input("Enter id:"))
    h = int(input("Enter h:"))
    hp = int(input("Enter hp:"))
    ov = 0
    if h > 40:
        ov = 0.5 *(h - 40) * hp
    p = ov + hp * h
    print("id = ",id," ov = ",ov, " p = ", p)
    i = i + 1
```

| متغیر | هدف           |
|-------|---------------|
| n     | تعداد کارکنان |
| id    | شماره کارمندی |
| h     | ساعت کارکرد   |
| hp    | دستمزد ساعتی  |
| ov    | اضافه‌کاری    |
| i     | شمارنده       |
| p     | حقوق          |

دستور اول، با پیغام مناسب n را می‌خواند، دستور دوم، i را برابر ۱ قرار می‌دهد، دستور سوم، یک حلقه while ایجاد می‌کند و تا زمانی که i کوچک‌تر یا مساوی n باشد، اعمال زیر را انجام می‌دهد:

- با پیغام‌های مناسب id (شماره کارمندی)، h (ساعت کارکرد)، hp (دستمزد ساعتی) را می‌خواند.
- ov (اضافه‌کاری) را برابر صفر قرار می‌دهد.
- اگر h بزرگ‌تر از ۴۰ باشد (یعنی اضافه‌کار داشته است)، آنگاه مبلغ ov با فرمول بیان شده (یعنی  $0.5 * (h - 40) * hp$ ) حساب می‌کند.
- مبلغ حقوق کارمند (یعنی p) را که برابر است با اضافه‌کار (ov) به‌علاوه ساعت کارکرد (h) ضرب در مبلغ هر ساعت hp است، به دست می‌آورد.
- با پیغام مناسب id، ov و p را نمایش می‌دهد.
- یکی به i اضافه می‌کند تا اطلاعات کارمندی بعدی را دریافت کرده، حقوقش را محاسبه کند.

خروجی برنامه:

```
Enter n:2
Enter id:1
Enter h:100
Enter hp:9000
id = 1 ov = 270000.0 p = 1170000.0
Enter id:2
Enter h:37
Enter hp:8000
id = 2 ov = 0 p = 296000
```

۹. یکی از مسائل مهم شرکت‌ها و ادارات پیش‌بینی بودجه چند سال بعد است. برنامه‌ای که قیمت کالا (ارزش پروژه) و نرخ تورم کالا را گرفته، قیمت کالا را برای چند سال بعد پیش‌بینی می‌نماید (برنامه تعداد سال‌هایی که قیمت کالا باید پیش‌بینی شود، از کاربر دریافت می‌نماید).

| متغیر | هدف                                 |
|-------|-------------------------------------|
| n     | سال پیش‌بینی بودجه                  |
| p     | ارزش کالا در سال جاری و سال‌های بعد |
| i     | شمارنده                             |



## ساختار تصمیم و حلقه تکرار ۴۱

```
p=int(input("Enter p:"))
n = int(input("Enter n:"))
inc = int(input("Enter inc:"))
print("Year      Price")
for i in range(1, n+1):
    p = p + (p * inc / 100)
    print( i,"      ", p)
```

|                    |     |
|--------------------|-----|
| نرخ تورم (به درصد) | inc |
|--------------------|-----|

دستور اول، قیمت کالا در سال فعلی را خوانده، در p قرار می‌دهد، دستور دوم، تعداد سال‌هایی که باید بودجه آن پیش‌بینی شود را خوانده، در n قرار می‌دهد، دستور سوم، نرخ تورم را خوانده، در inc قرار می‌دهد، دستور چهارم، یک عنوان را نمایش می‌دهد (عنوان "Year Price")، دستور پنجم، یک حلقه ایجاد می‌کند که از ۱ تا n می‌شمارد تا ارزش کالا را در n سال بعد حساب کند و نمایش دهد. در این حلقه، ابتدا، p (ارزش کالا در سال بعد) ۱ برابر p (ارزش کالا در سال فعلی) به علاوه p (ارزش کالا در سال فعلی) \* inc (نرخ تورم) تقسیم بر ۱۰۰ می‌کند. سپس i و p را نمایش می‌دهد.

خروجی برنامه:

```
Enter p:100000
Enter n:5
Enter inc:20
Year      Price
1         120000.0
2         144000.0
3         172800.0
4         207360.0
5         248832.0
```

۱۰. برنامه‌ای که مجموع ده جمله اول سری زیر را محاسبه و چاپ می‌کند.

$$s = \frac{1}{x} - \frac{1}{x + 2x^2} + \frac{1}{x + 2x^2 + 3x^3} - \frac{1}{x + 2x^2 + 3x^3 + 4x^4} + \dots$$

```
s = 1
pow = 1.0
sum = 0.0
sum1 = 0.0
x = int(input("Enter x:"))
for i in range(1, 11):
    pow = pow * x
    sum1 = sum1 + i * pow
    sum = sum + s * 1.0 / sum1
    s = -s
print("Sum is ", sum)
```

| متغیر | هدف                  |
|-------|----------------------|
| x     | عدد خوانده شده       |
| i     | شمارنده              |
| pow   | توان‌های x           |
| sum   | مجموع                |
| sum1  | مجموع مخرج           |
| s     | علامت یک‌درمیان منفی |

دستور اول s (علامت که یکی در میان مثبت و منفی می‌شود) را برابر ۱، (چون جمله اول سری دارای علامت + است) قرار می‌دهد، دستور دوم، توان x (یعنی pow) را برابر ۱ قرار می‌دهد، دستور سوم، sum (مجموع کل سری) را برابر ۰،۰ قرار می‌دهد، دستور پنجم، با پیغام مناسب x را می‌خواند، دستور ششم، یک حلقه با شمارنده i ایجاد می‌کند که از ۱ تا ۱۰ شمارش می‌کند و در داخل حلقه، اعمال زیر را انجام می‌دهد:

۱. pow را در x ضرب می‌کند و در pow قرار می‌دهد. این جمله برای محاسبه توان‌های x به کار می‌رود. یعنی، وقتی i برابر ۱ باشد، pow برابر x است. اما، i برابر ۲ باشد، pow برابر x<sup>۲</sup> می‌شود و همین‌طور ادامه می‌یابد.

۲. sum1 برابر sum + pow \* i می‌شود تا در هر مرحله مخرج کسر حساب گردد.

۳. sum برابر sum قبلی به علاوه s (علامت) ضرب در ۰، ۱ (تقسیم اعشاری) تقسیم بر ۱ sum می‌شود.
۴. s برابر -s می‌شود تا در گام‌های بعدی علامت تغییر یابد (یعنی، علامت s به ترتیب مقادیر ۱، -۱، ۱، -۱ و ... را در گام‌های مختلف حلقه می‌پذیرد). در پایان، با پیغام مناسب sum را نمایش می‌دهد.

خروجی برنامه:

```
Enter x:5
Sum is 0.18384874697896247
```

۱۱. برنامه‌ای که مجموعه‌ای از اعداد دوتایی صحیح را خوانده، حاصل ضرب آن‌ها را بدون استفاده از عملگر ضرب نمایش می‌دهد. چنانچه کاربر به جای دو عدد، صفر وارد کند، برنامه خاتمه یابد.

```
while True:
    sum = 0
    x = int(input("Enter x:"))
    y = int(input("Enter y:"))
    if (x == 0 and y == 0):
        break
    temp = y
    if (y < 0):
        temp = -y;
    for i in range(1, temp+1):
        sum = sum + x
    if (y < 0):
        sum = -sum
    print(x, " * ", y, " = ", sum)
```

| متغیر | هدف              |
|-------|------------------|
| x     | مقدار اول        |
| y     | مقدار دوم        |
| i     | شمارنده          |
| sum   | مجموع (حاصل ضرب) |
| temp  | کمکی             |

دستور اول، یک حلقه while بی‌نهایت ایجاد می‌کند. در این حلقه، اعمال زیر انجام می‌گردد:

۱. متغیر sum را تعریف کرده، مقدار ۰ را به آن تخصیص می‌دهد.
۲. با دو پیغام x و y را می‌خواند.
۳. اگر x و y برابر ۰ باشند، با دستور break حلقه while را خاتمه می‌دهد.
۴. y را در temp قرار می‌دهد تا مقدار y از دست نرود.
۵. اگر y کوچک‌تر از صفر باشد، -y را در temp قرار خواهد داد.
۶. با یک حلقه تکرار از ۱ تا temp می‌شمارد و در داخل حلقه sum را با x جمع کرده، در sum قرار می‌دهد تا sum حاصل ضرب شود.
۷. اگر y منفی باشد، -sum را در sum قرار می‌دهد تا نتیجه ضرب در یک منفی ضرب شود.
۸. با پیغام مناسب sum را نمایش می‌دهد.

خروجی برنامه:

```
Enter x:10
Enter y:-8
10 * -8 = -80
Enter x:-8
Enter y:-10
-8 * -10 = 80
Enter x:0
Enter y:0
```

۱۲. برنامه‌ای که عددی را خوانده، مشخص می‌کند آیا عدد متقارن است یا خیر. نمونه‌های از اعداد متقارن عبارت‌اند از: ۱۲۴۲۱ و ۳۲۴۲۳. (توضیح: چنانچه عددی برابر با مغلوبش باشد، متقارن است. پس، ابتدا مغلوب عدد را محاسبه کرده، با خودش مقایسه می‌کنیم.)

### ساختار تصمیم و حلقه تکرار ۴۳

```

pow = 10
sum = 0
num = int(input("Enter a number:"))
temp = num
while temp > 0:
    sum = (pow * sum) + temp % 10
    temp = temp // 10
if (sum == num):
    print("Yes")
else:
    print("No")

```

| متغیر | هدف                    |
|-------|------------------------|
| num   | عدد وارد شده           |
| pow   | ثابت ۱۰                |
| temp  | کمکی برای محاسبه مغلوب |
| sum   | مجموع (مغلوب عدد)      |

دستور اول، pow را برابر ۱۰ قرار می‌دهد تا در هر مرحله pow ضرب در sum شود، با رقم یکان جمع گردد تا مغلوب عدد محاسبه شود، دستور دوم sum (مغلوب عدد) را ۰ قرار می‌دهد، دستور سوم، با پیغام مناسب، یک عدد را خوانده در num قرار می‌دهد، دستور چهارم، num را در temp قرار می‌دهد تا در زمانی که ارقام عدد را جدا می‌کند، num (عدد خوانده شده) را از دست ندهد، دستور پنجم، یک حلقه while ایجاد می‌کند. این حلقه تا زمانی که temp بزرگ‌تر از صفر باشد، اعمال زیر را انجام می‌دهد (این حلقه برای به دست آوردن مغلوب num از temp استفاده می‌کند):

۱. sum را در pow ضرب کرده، با رقم یکان temp (temp//۱۰) جمع می‌کند و در sum قرار می‌دهد تا بتواند در هر مرحله مغلوب عدد را حساب کند.
۲. رقم یکان temp را حذف می‌کند (temp//=۱۰).

دستور بعدی، اگر sum برابر num باشد (یعنی، مغلوب عدد برابر خود عدد است)، آنگاه "Yes" را نمایش می‌دهد، وگرنه "No" را نمایش خواهد داد.

خروجی برنامه:

```

Enter a number: ۱۲۲۲۱
Yes

```

### ۱۳. برنامه‌ای که m و n دو عدد صحیح و مثبت را خوانده، با استفاده از عملگر +، m<sup>n</sup> را محاسبه و چاپ می‌نماید.

```

m = int(input("Enter n:"))
n = int(input("Enter m:"))
sum = 0
temp = m
for i in range(1, n):
    sum = 0
    for j in range(1, m + 1):
        sum = sum + temp
    temp = sum
print(m, " ^ ", n, " = ", sum)

```

| متغیر | هدف                                      |
|-------|--|
| m     | پایه                                     |
| n     | توان                                     |
| i     | شماره ۱ تا n-۱                           |
| j     | شمارنده ۱ تا m                           |
| sum   | مجموع (در پایان مقدار توان قرار می‌گیرد) |
| temp  | متغیر کمکی برای نگهداری مجموع هر دوره    |

دستورات اول و دوم، با پیغام‌های مناسب دو عدد را خوانده، در m و n قرار می‌دهند، دستور سوم، sum را صفر قرار می‌دهد (چون توان را با عملگر جمع محاسبه می‌کنیم، متغیر sum را در نظر گرفتیم)، دستور چهارم، m را در temp قرار می‌دهد (temp، عددی است که هر مرحله باید جمع شود، در اولین مرحله m است ولی در مراحل بعدی، باید حاصل جمع مراحل قبلی را داشته باشد)، دستور پنجم، یک حلقه از ۱ تا n-۱ ایجاد می‌کند تا بتواند توان را از طریق جمع محاسبه کند. در این حلقه اعمال زیر انجام می‌شود:

۱. sum (همان مقدار توان) را برابر ۰ قرار می‌دهد.

۲. در یک حلقه از ۱ تا m، sum را با temp جمع کرده، در sum قرار می‌دهد.
  ۳. sum را در temp قرار می‌دهد تا در مرحله بعد، این مجموع m بار با خودش جمع شود.
- در پایان، با پیغام مناسب، sum را نمایش می‌دهد.

خروجی برنامه:

```
Enter n:2
Enter m:10
2 ^ 10 = 1024
```

۱۴. برنامه‌ای که تعدادی عدد را خوانده، هر یک از اعداد را که بر ۹ بخش‌پذیر باشد، در خروجی چاپ می‌کند (از روش مجموع ارقام استفاده کنید).

```
n = int(input("Enter n:"))
for i in range(1, n + 1):
    num = int(input("Enter num:"))
    temp = num
    sum = 0
    while (temp > 0):
        sum = sum + temp % 10
        temp //= 10
    if (sum % 9 == 0):
        print(num)
```

| متغیر | هدف           |
|-------|---------------|
| n     | تعداد اعداد   |
| i     | شمارنده       |
| sum   | مجموع ارقام   |
| temp  | کمکی          |
| num   | اعداد دریافتی |

دستور اول، با پیغام مناسب n را می‌خواند، دستور دوم، یک حلقه ایجاد می‌کند که از ۱ تا n می‌شمارد تا n عدد را بخواند. در داخل این حلقه اعمال زیر انجام می‌گردد:

۱. با پیغام مناسب یک عدد را خوانده، در num قرار می‌دهد.
۲. num را در temp قرار می‌دهد تا عدد خوانده شده را از دست ندهد.
۳. sum (مجموع ارقام) را برابر ۰ قرار می‌دهد.
۴. با استفاده از یک while، مجموع ارقام temp (همان num) را حساب می‌کند. برای این منظور، در هر مرحله sum را با رقم یکان temp ( $temp \% 10$ ) جمع کرده و رقم یکان temp ( $temp // 10$ ) را حذف می‌کند.
۵. اگر باقی‌مانده sum به ۹ برابر ۰ باشد، پس num بر ۹ بخش‌پذیر است، آنگاه num را نمایش می‌دهد.

خروجی برنامه:

```
Enter n:4
Enter num:18
18
Enter num:8
Enter num:123
Enter num:981
981
```

۱۵. برنامه‌ای که اطلاعات تعدادی کارمند از قبیل شماره کارمندی، سال استخدام، کد جنسیت، (۰: زن، ۱: مرد) پایه استخدام (۹-۱) و حقوق را خوانده، موارد زیر را در خروجی چاپ می‌کند.

الف. تعداد کارمندان مرد      ب. تعداد کارکنان زن      ج. میانگین حقوق به تفکیک زن و مرد

| متغیر | هدف           |
|-------|---------------|
| id    | شماره کارمندی |
| date  | سال استخدام   |
| m_w   | کد جنسیت      |

```

n = int(input("Enter n:"))
base6 = 0
t_w = 0
t_m = 0
sumpay_m = 0
sumpay_w = 0
for i in range(1, n+1):
    id = int (input ("Enter a id:"))
    date = int (input ("Enter a date:"))
    m_w = int (input ("Enter a m_w(0|1):"))
    base1 =int (input("Enter base:"))
    pay = int (input ("Enter pay:"))
    if base1 == 6:
        base6 = base6 + 1
    if (m_w == 0):
        sumpay_w = sumpay_w + pay
        t_w = t_w + 1
    if (m_w == 1):
        sumpay_m = sumpay_m + pay
        t_m = t_m + 1
print("***** result ***** \n Number of base six is ", base6)
print("Number of woman is ", t_w)
if (t_w > 0):
    print("Average(woman) pay is ", 1.0 * sumpay_w / t_w)
if (t_m > 0):
    print("Average(man) pay is ", 1.0 * sumpay_m / t_m)

```

|                         |          |
|-------------------------|----------|
| پایه استخدام            | base۱    |
| حقوق کارمند             | pay      |
| تعداد کارمندان پایه ۶   | base۶    |
| تعداد کارمندان زن       | t_w      |
| تعداد کارمندان مرد      | t_m      |
| تعداد کارمندان          | n        |
| حقوق کارمندان           | pay      |
| مجموع حقوق کارمندان زن  | sumpay_w |
| مجموع حقوق کارمندان مرد | sumpay_m |
| شمارنده                 | i        |

دستور اول، n (تعداد کارمندان) را می‌خوانده، دستورات دوم تا ششم، متغیرهای base۶ (تعداد کارمندان پایه ۶)، t\_w (تعداد کارمند زن)، t\_m (تعداد کارمندان مرد)، sumpay\_m (مجموع حقوق کارمندان مرد) و sumpay\_w (مجموع حقوق کارمندان زن) را برابر ۰ قرار می‌دهند. دستور هفتم، یک حلقه از ۱ تا n ایجاد می‌کند تا اطلاعات n کارمند را بخواند و پردازش‌های لازم را بر روی آن انجام دهد. در این حلقه اعمال زیر انجام می‌شود:

- با پیغام مناسب، شماره کارمندی، تاریخ استخدام، جنسیت، پایه و حقوق را خوانده، به ترتیب در متغیرهای id، date، m\_w، base۱ و pay قرار می‌دهد.
  - اگر مقدار base۱ (پایه) برابر ۶ باشد، به تعداد کارمندان پایه ۶ یکی اضافه می‌کند.
  - اگر m\_w برابر صفر باشد (یعنی، کارمند زن است)، آنگاه حقوق (pay) را با sumpay\_w (مجموع حقوق زنان) جمع کرده، در sumpay\_w قرار می‌دهد و به تعداد کارمندان زن (t\_w) یک واحد اضافه می‌کند.
  - اگر m\_w برابر یک باشد (یعنی، کارمند مرد است)، آنگاه حقوق را با sumpay\_m (مجموع حقوق مردان) جمع کرده، در sumpay\_m قرار می‌دهد و به تعداد کارمندان مرد t\_m یکی اضافه می‌کند.
- در پایان حلقه، ابتدا پیغام "\*\*\*\*\* result \*\*\*\*\*" را نمایش می‌دهد و به سطر بعد می‌رود، سپس پیغام "Number of base six is" را به همراه مقدار base۶ (تعداد کارمندان پایه ۶) نشان می‌دهد، در ادامه، اگر t\_w بزرگ‌تر از صفر باشد (یعنی، کارمند زنی وجود داشته باشد)، آنگاه میانگین حقوق کارمندان زن

(یعنی  $sum/t_w * 10^1$ ) را با پیغام مناسب نمایش می‌دهد. در پایان، اگر  $t_m$  بزرگ‌تر از صفر باشد، میانگین حقوق کارمندان مرد را نیز با پیغام مناسب نمایش خواهد داد.

خروجی برنامه:

```
Enter n:3
Enter a id:1
Enter a date:1396
Enter a m_w(0|1):1
Enter base:6
Enter pay:5000000
Enter a id:2
Enter a date:1380
Enter a m_w(0|1):0
Enter base:4
Enter pay:4000000
Enter a id:3
Enter a date:1357
Enter a m_w(0|1):1
Enter base:6
Enter pay:6000000
***** result *****
Number of base six is 2
Number of woman is 1
Average(woman) pay is 4000000.0
Average(man) pay is 5500000.0
```

۱۶. برنامه‌ای که رشته‌ای را خوانده، تعداد حروف صدادار رشته را شمارش می‌کند (حروف صدادار عبارت‌اند از: `e`، `i`، `u`، `o`، `&`؛). هدف برنامه آشنایی بیشتر با پیمایش رشته با for است.

```
s = input("Enter a string:")
sound = "aAeEiIoOuU"
count = 0
for i in s:
    if i in sound:
        count = count+1
print("Count is ", count)
```

| متغیر | هدف  |
|-------|--|
| S     | رشته خوانده شده                              |
| sound | رشته‌ای که تمام حروف صدادار در آن قرار دارند |
| count | تعداد حروف صدادار                            |
| i     | هر کاراکتر رشته S را پیمایش می‌کند           |

دستور اول، یک رشته را خوانده، در S قرار می‌دهد، دستور دوم، تمام کاراکترهایی که حروف صدادار هستند را در رشته sound قرار می‌دهد، دستور سوم، تعداد کاراکترهای صدادار (count) را برابر ۰ قرار می‌دهد، دستور چهارم، در یک حلقه تمام کاراکترهای رشته S را پیمایش کرده، هر کاراکتر آن را در i قرار می‌دهد و هر بار در حلقه، اگر i در sound باشد (یعنی، i یک کاراکتر صدادار باشد)، به count یک واحد اضافه می‌کند. در پایان حلقه، مقدار count را با پیغام مناسب نمایش می‌دهد.

خروجی برنامه:

```
Enter a string:Python language
Count is  ۵
```

۱۷. برنامه‌ای که خروجی زیر را ایجاد می‌کند:

```
7      6      5      4      3      2      1
6      5      4      3      2      1
5      4      3      2      1
4      3      2      1
3      2      1
2      1
1
```

## ساختار تصمیم و حلقه تکرار ۴۷

```
for i in range(7, 0, -1):
    print ()
    for j in range(i, 0, -1):
        print(j, end = '\t')
```

| متغیر | هدف                        |
|-------|----------------------------|
| i     | شمارنده حلقه خارجی (۷..۱)  |
| j     | شمارنده حلقه داخلی (۱...i) |

دستور اول، یک حلقه ایجاد می کند که شمارنده i آن از ۷ تا ۱ به صورت نزولی مقدار می گیرد. در داخل این حلقه ابتدا، کنترل چاپ را به سطر بعدی انتقال می دهد، سپس یک حلقه دیگر با شمارنده j ایجاد می کند که از i تا ۱ به صورت نزولی مقدار می گیرد و در داخل این حلقه مقدار j را چاپ کرده، کنترل چاپ را به Tab بعدی انتقال می دهد.

**۱۸. بازی سنگ، کاغذ و قیچی دو بازیکن دارد. بازیکن اول می تواند قیچی (مقدار ۱)، سنگ (مقدار ۲) و یا کاغذ (مقدار ۳) را انتخاب کند و بازیکن دوم همین طور. اگر انتخاب دو بازیکن یکسان باشد، مساوی اند. اگر متفاوت باشد، یک نفر برنده است و برنده یک امتیاز را کسب می کند. برای مثال:**

✚ اگر بازیکن اول قیچی و بازیکن دوم کاغذ را انتخاب کند، بازیکن اول برنده است. چون قیچی کاغذ را می برد.  
 ✚ اگر بازیکن اول قیچی و بازیکن دوم سنگ را انتخاب کند، بازیکن دوم برنده است. زیرا سنگ قیچی را می شکند.  
 ✚ اگر بازیکن اول سنگ و بازیکن دوم کاغذ را انتخاب کند، بازیکن دوم برنده است. زیرا کاغذ دور سنگ می پیچد.  
 برنامه ای که این مسابقه را ۵ دور بین دو بازیکن انجام می دهد و برنده را نمایش می دهد (بازیکن ها از انتخاب هم دیگر خبر ندارند).

```
score1 = 0
score2 = 0
for i in range(1, 6):
    player1 = int(input("Player 1 Please Enter 1:(scissors)\
        , 2:(stone) , 3:(paper):"))
    player2 = int(input("Player 2 Please Enter 1:(scissors)\
        , 2:(stone) , 3:(paper):"))
    if ((player1 == 1 and player2 == 3) or (player1 == 3 and \
        player2 == 2) or (player1 == 2 and player2 == 1)):
        score1 = score1 + 1
    elif ((player2 == 1 and player1 == 3) or (player2 == 3 \
        and player1 == 2) or (player2 == 2 and player1 == 1)):
        score2 = score2 + 1;
    print("\nScore for player1 is:", score1)
    print("Score for player2 is:", score2)
```

دستورات اول و دوم به ترتیب امتیاز بازیکن اول (score۱) و امتیاز بازیکن دوم (score۲) را صفر می کنند، دستور سوم، یک حلقه ایجاد می کند که از ۱ تا ۵ شمارش می کند، در داخل حلقه اعمال زیر را انجام می دهد:

- با پیام مناسب انتخاب بازیکن اول را دریافت کرده، در player۱ قرار می دهد.
- با پیام مناسب انتخاب بازیکن دوم را دریافت کرده، در player۲ قرار می دهد.
- اگر بازیکن اول قیچی و بازیکن دوم کاغذ (player۱==۱ and player۲ == ۳) یا بازیکن اول کاغذ و بازیکن دوم، سنگ (player۱==۳ and player۲ == ۲) یا بازیکن اول سنگ و بازیکن دوم قیچی (player۱==۲ and player۲ == ۱) را انتخاب کرده باشد، آنگاه به امتیاز بازیکن اول (score۱) یک واحد اضافه می شود.
- اگر بازیکن دوم قیچی، بازیکن اول کاغذ یا بازیکن دوم کاغذ و بازیکن اول سنگ یا بازیکن اول سنگ و بازیکن دوم قیچی را انتخاب کرده باشد، به امتیاز بازیکن دوم (score۲) یک واحد اضافه می شود.

در پایان، با پیام های مناسب امتیاز بازیکن اول (score۱) و امتیاز بازیکن دوم (score۲) را نمایش می دهد.

**خروجی برنامه:**

```
Player 1 Please Enter 1:(scissors), 2:(stone) , 3:(paper):1
Player 2 Please Enter 1:(scissors), 2:(stone) , 3:(paper):2
Player 1 Please Enter 1:(scissors), 2:(stone) , 3:(paper):2
Player 2 Please Enter 1:(scissors), 2:(stone) , 3:(paper):3
Player 1 Please Enter 1:(scissors), 2:(stone) , 3:(paper):1
Player 2 Please Enter 1:(scissors), 2:(stone) , 3:(paper):3
Player 1 Please Enter 1:(scissors), 2:(stone) , 3:(paper):3
Player 2 Please Enter 1:(scissors), 2:(stone) , 3:(paper):3
Player 1 Please Enter 1:(scissors), 2:(stone) , 3:(paper):2
Player 2 Please Enter 1:(scissors), 2:(stone) , 3:(paper):2
Player 1 Please Enter 1:(scissors), 2:(stone) , 3:(paper):1
Player 2 Please Enter 1:(scissors), 2:(stone) , 3:(paper):2
```

```
Score for player1 is: 2
Score for player2 is: 3
```

۱۹. برنامه‌ای که تمام اعداد بین ۱۰۰۰ تا ۱۱۰۰ که بر ۹ بخش‌پذیرند را نمایش می‌دهد (عددی بر ۹ بخش‌پذیر است که مجموع ارقام آن بر ۹ قابل قسمت باشد).

```
print("Result is:")
for i in range(1000, 1101):
    sum = 0
    n = i
    while (n > 0):
        sum = sum + (n % 10)
        n //= 10;
    if (sum % 9 == 0):
        print(i,end = '\t')
```

| متغیر | هدف                     |
|-------|-------------------------|
| i     | شمارنده از ۱۰۰۰ تا ۱۱۰۰ |
| n     | عدد مربوط               |
| sum   | مجموع ارقام             |

این برنامه ابتدا متغیرهای موردنیاز را تعریف کرده، سپس پیام Result is را چاپ می‌کند. در ادامه با استفاده از یک حلقه for از ۱۰۰۰ تا ۱۱۰۰ شمارش می‌کند و در داخل این حلقه، مقدار i را در n قرار می‌دهد (مقدار i از دست نرود). سپس ارقام n را جدا کرده، با یکدیگر جمع می‌نماید (حلقه while). در پایان حلقه while، اگر مقدار sum بر ۹ بخش‌پذیر باشد، i (شمارنده عدد) را نمایش می‌دهد.

خروجی برنامه:

```
Result is:
1008 1017 1026 1035 1044 1053 1062 1071 1080 1089 1098
```

۲۰. برنامه‌ای که کاراکتری را که نشان‌دهنده رنگی می‌باشد، از ورودی خوانده، می‌گوید چه رنگی را انتخاب کردید. مثلاً اگر 'r' یا R را وارد کردید، برنامه به شما می‌گوید دوست دارید رنگ قرمز (Red) را انتخاب کنید. برای خروج کاربرد باید e یا E را وارد کنید.

```
loop = 1
while loop == 1:
    ch = input("Enter a char:");
    if ch[0] == 'w' or ch[0] == 'W':
        print ("You love White color")
    elif ch[0] == 'r' or ch[0] == 'R':
        print("You love Red color")
    elif ch[0] == 'y' or ch[0] == 'Y':
        print("You love Yellow color")
    elif ch[0] == 'b' or ch[0] == 'B':
        print ("You love Blue color")
    elif ch[0] == 'g' or ch[0] == 'G':
        print("You love Green color")
    elif ch[0] == 'e' or ch[0] == 'E':
        loop = 0;
    else:
        print("No color chosen")
```

| متغیر | هدف            |
|-------|----------------|
| ch    | کاراکتر ورودی  |
| loop  | شرط تکرار حلقه |



## ساختار تصمیم و حلقه تکرار ۴۹

دستور اول، متغیر loop را با مقدار اولیه ۱ تعریف می‌کند، دستور دوم، تازمانی که مقدار loop برابر ۱ باشد، یک حلقه while را تکرار می‌کند و در داخل این حلقه اعمال زیر را انجام می‌دهد:

۱. ابتدا با یک پیغام مناسب یک رشته را خوانده، در ch قرار می‌دهد.
۲. اگر کاراکتر اول ch (ch[۰]) برابر 'w' یا 'W' باشد، پیغام "You love White color" را نمایش می‌دهد.
۳. وگرنه، اگر کاراکتر اول ch برابر 'r' یا 'R' باشد، پیغام "You love Red color" را نمایش می‌دهد.
۴. وگرنه، اگر کاراکتر اول ch برابر 'y' یا 'Y' باشد، پیغام "You love Yellow color" را نمایش می‌دهد.
۵. وگرنه، اگر کاراکتر اول ch برابر 'b' یا 'B' باشد، پیغام "You love Blue color" را نمایش می‌دهد.
۶. وگرنه، اگر کاراکتر اول ch برابر 'g' یا 'G' باشد، پیغام "You love Green color" را نمایش می‌دهد.
۷. وگرنه، اگر کاراکتر اول ch برابر 'e' یا 'E' باشد، مقدار loop را برابر ۰ می‌دهد تا حلقه while خاتمه یابد.

وگرنه پیغام "No color choosen" را نمایش می‌دهد.

خروجی برنامه:

```
Enter a char:y
You love Yellow color
Enter a char:R
You love Red color
Enter a char:E
```

۲۱. برنامه‌ای که دو عدد مثبت را از ورودی خوانده، آنگاه عدد بزرگ‌تر را به روش تفریق بر عدد کوچک‌تر تقسیم می‌کند.

```
i = 0
x = int(input("Enter x:"))
y = int(input("Enter y:"))
if x > y:
    temp = x
    r = y
else:
    temp = y
    r = x
while temp >= r:
    temp -= r
    i = i + 1
if x > y:
    print(x, " / ", y, " = ", i)
else:
    print(y, " / ", x, " = ", i)
```

| متغیر | هدف                         |
|-------|-----------------------------|
| i     | تعداد تفریق‌ها (حاصل تقسیم) |
| x     | عدد اول                     |
| y     | عدد دوم                     |
| temp  | عدد بزرگ‌تر                 |
| r     | عدد کوچک‌تر                 |

دستور اول، حاصل تقسیم (i) را برابر صفر قرار می‌دهد، دستورات دوم و سوم، با پیغام‌های مناسب دو عدد را خوانده، در x و y قرار می‌دهند، دستور چهارم، اگر x بزرگ‌تر از y باشد، temp (عدد بزرگ‌تر) را برابر x قرار می‌دهد و عدد کوچک‌تر (r) را برابر y قرار می‌دهد، وگرنه، y را در temp و x را در r قرار می‌دهد، دستور while تا زمانی که temp بزرگ‌تر یا مساوی r باشد (یعنی، از temp بتوان یک دسته r تایی جدا نمود)، ابتدا از temp واحد کم می‌کند و به حاصل تقسیم (یعنی، i) یک واحد اضافه می‌کند.