

به نام خدا

گزارش تمرین پنجم

9623088

سوال 1:

در این سوال برای استفاده نکردن از حلقه از توابع و کتابخانه هایی استفاده شده است. در مورد اجرای موازی و قسمت آخر، کتابخانه یاد شده شناخته نشد.

سوال 2:

به هر تعداد که اولش آرگومان ساده بدهیم، یعنی قبل از `*args, **keywords` متغیر ساده داریم که مقادیر در آن ها قرار می گیرد.

از آن به بعد تا جایی که تنها یک مقدار بدهیم داریم از `*args`

استفاده می کنیم و مقادیر در یک `Tuple` ریخته می شوند. سپس اگر چیزی را

مساوی چیز دیگری قرار دهیم، این مقادیر به شکل یک `Value, key`

در یک `Dictionary` ذخیره خواهند شد.

سوال 3:

الف:

`A0= { 'a':'1' , 'b':'2' , 'c':'3' , 'd':'4' , 'e':'5' }`

`A1= range(10)`

$A2 = []$

$A3 = ['1', '2', '3', '4', '5']$

$A4 = [[0,0], [1,1],$

$[2,4], [3,9], [4,16], [5,25], [6,36], [7,49], [8,64], [9,81]]$

در $A0$ ابتدا دو توپل دریافتی به هم متصل شده و سپس از ترکیب دو دسته توپل دیکشنری تولید شده است.

در $A1$ مقادیر 0 تا 9 ذخیره خواهند شد اما چون واضحا برابر لیست یا توپل خاصی قرار داده نشده مقادیر درونش هستند و با چاپ آن اتفاق خاصی نمی افتد. در $A2$ یک List comprehension داریم که از مقادیر درون $A1$ هم استفاده شده.

در $A3$ با کمک List comprehension یک لیست مرتب شده می بینیم. در $A4$ هم با کمک List comprehension یک لیست از لیستهای دیگر تحویل می گیریم.

ب):

کد را به کمک `globals()` می نویسیم. که تمام متغیرهای برنامه از جمله آنهایی که ما نیازشان داریم را در خودش دارد. چون دایما در حال تغییر است از کپی آن استفاده خواهیم کرد. با کمک نام متغیرها آنهایی که می خواهیم پیدا می کنیم و

کدشان را اجرا می کنیم.

سوال 4:

کاملا مطابق صورت سوال نوشته شده.

سوال 5:

کد این سوال طوری نوشته شده که دائما ورودی می گیرد و برای اتمام این چرخه میتوان از Tab استفاده کرد. ورودی ها دریافت می شوند و در صورت تطابق با دستورات ذکر شده در صورت سوال برای جدا کردن ورودی توابع و آرگومان های مورد نیاز عبارت ورودی تجزیه می شود. (با کمک دستور Split(',') به این صورت که عبارت قبل "،" المان اول لیست و عبارت بعد آن المان دوم یک لیست می شوند. حال چون دستور را می دانیم چه بوده ، می توان آدرسها و اسامی (آرگومانهای ورودی توابع) را جدا کرد و به توابع داد. با os.path.join() اسم و آدرس را به هم متصل کرده و وجودشان را چک می کنیم و اقدام لازمه را انجام میدهیم.

برای تابع find از os.walk() استفاده می کنیم که فایل های یک پوشه را می گردد و پیدا می کند.

سوال 6:

سوال هیچ نکته جدیدی نداشته و مطابق صورت سوال نوشته شده است.

سوال 7:

ورودی دریافت شده، 6 تا 6 جدا می شود و حلقه در آن جستجو خواهد کرد.
چون قرار است که جایگاه عدد هم مضرب 6 باشد.
سپس چک می شود که آیا خود عدد هم مضرب 6 هست یا نه.
(از تابع `eval()` چون خروجی تابع `input()` از نوع رشته خواهد بود.)
بعد مقدار لیست حاوی این اعداد مرتب می شود تا یک لیست صعودی داشته
باشیم و در نهایت هم برای اینکه در یک خط چاپ شوند گفته شده آخرشان از
فاصله استفاده شود.

Git : <https://github.com/k-ghodsifar/AP-HW5.git>