



سوال ۱	سه مورد از اجزاء اصلی در ساختار کامپیوتر را نام ببرید و کارکرد هر یک را مختصر توضیح دهید.
سوال ۲	تفاوت الگوریتم و برنامه چیست؟ با ذکر مثال بیان کنید.
سوال ۳	پس از اجرای خط زیر، خروجی چه خواهد بود؟ راه حل خود را مختصر توضیح دهید. $\text{result} = 5 * 4 ** 3 // 4 + (6 - 8   0)$
سوال ۴	دنباله‌ای از اعداد را در نظر بگیرید که در آن هر عدد، حاصل ضرب ارقام عدد قبلی است. می‌خواهیم برنامه‌ای بنویسیم که این دنباله را تولید کند. تابعی بنویسید که یک عدد را از کاربر دریافت کرده و آن را به عنوان اولین عدد دنباله در نظر بگیرد. این تابع باید تولید اعداد دنباله را تا جایی ادامه دهد که حاصل ضرب ارقام عدد فعلی، یک عدد اول شود. پس از اتمام تولید این اعداد، تابع شما باید دنباله‌ی تولیدشده و عدد اولی را که در پایان حاصل شده، چاپ نماید. توجه: اگر حاصل ضرب ارقام عدد قبلی، تک رقمی شد، عدد تک‌رقمی را با ۲۰ جمع نمایید.
سوال ۵	عدد 10213 در مبنای ۴ نوشته شده است. نمایش آن در مبنای ۹ را بنویسید.
سوال ۶	خروجی تکه کد زیر چیست؟ راه حل خود را مختصر توضیح دهید. <pre>def modify_objects(obj1, obj2):     obj1[0] = 'X'     obj1.append(obj2[0])     obj2 = obj2 + 'Y'  my_list = [1, 2, 3, ['A', 'B']] my_string = 'ABC'  modify_objects(my_list, my_string)  print("List:", my_list) print("String:", my_string)</pre>
سوال ۷	با مفهوم ترکیب توابع در گذشته آشنا شده‌اید. برای مثال، اگر داشته باشیم $f(x) = x^{**2}$ $g(x) = x^{**3}$ $\text{fog}(x) = f(g(x)) = (x^{**3})^{**2} = x^{**6}$ آنگاه می‌توان نوشت: فرض کنید یک عدد ثابت مثل $x$ داده شده است. تابعی بنویسید که با دریافت تعداد دلخواهی تابع به عنوان ورودی، از انتها به ابتدا آن‌ها را روی $x$ اثر دهد و در صورتی که حاصل اثر تابع فعلی، عدد قبلی را کوچک‌تر نماید، همان تابع قبلی را به جای تابع فعلی در نظر بگیرد.



سوال ۸	می‌خواهیم تابعی بنویسیم که رفتاری مشابه تابع zip را پیاده‌سازی کند. این تابع را custom_zip می‌نامیم. تابع custom_zip باید دو لیست با طول‌های مختلف را دریافت کرده و یک شیء تکرارکننده (iterable) ایجاد کند که تاپل‌های حاوی عناصر از لیست‌های ورودی تولید کند. برخلاف zip اصلی، تابع شما باید با طول‌های مختلف به خوبی کار کند و لیست بلندتر را برش داده تا به طول لیست کوتاه‌تر برسد.
سوال ۹	برنامه‌ای بنویسید که تعداد راه‌های مختلف برای ترکیب سکه‌ها جهت به‌دست آوردن یک مقدار مشخص را محاسبه کند. این برنامه باید مقدار مورد نظر و یک لیست از ارزش‌های مختلف سکه‌ها را به عنوان ورودی بگیرد و تعداد کل ترکیب‌های یکتا برای به‌دست آوردن این مقدار را خروجی دهد. راهنمایی: برای حل این سوال از برنامه‌ریزی پویا (Dynamic Programming) استفاده کنید.
سوال ۱۰	یک مک آدرس عبارتی به شکل زیر است: 01:23:45:67:89:ab عبارت منظمی بنویسید که مک آدرس‌های معتبر را شناسایی کند. یک مک آدرس معتبر باید با قوانین زیر سازگار باشد: ۱. مک آدرس شامل ۶ گروه از اعداد دورقمی در مبنای ۱۶ است. (یعنی شامل ارقام ۰ تا ۹ و حروف a-f یا A-F می‌شود). ۲. طول کل عبارت مک آدرس ۱۷ کاراکتر است. ۳. میان هر دو گروه یک علامت دو نقطه وجود دارد. توجه کنید که در ابتدای مک آدرس دو نقطه نداریم.
سوال ۱۱	فرض کنید یک رشته به نام s دارید که شامل کلماتی است که توسط کاراکتر فاصله جدا شده‌اند. با این حال، برخی از کلمات در رشته شامل کاراکترهای خاص هستند. وظیفه شما این است که هر کلمه را در رشته به صورت معکوس بچرخانید و در عین حال جایگاه کاراکترهای خاص داخل هر کلمه را حفظ کنید. یک تابع به نام reverse_words بنویسید که یک رشته را به عنوان ورودی بگیرد و یک رشته جدید با کلمات معکوس را با حفظ مکان کاراکترهای خاص برگرداند. قوانین مربوط به معکوس کردن کلمات با کاراکترهای خاص به شرح زیر است: یک "کلمه" به عنوان یک دنباله از کاراکترهای غیر فاصله تعریف می‌شود. کاراکترهای خاص (هر چیز به جز حروف و ارقام) باید در جایگاه اصلی خود درون هر کلمه باقی بمانند. رشته معکوس باید جایگاه اصلی کلمات و فاصله‌ها را حفظ کند. مثال: اگر فراخوانی زیر انجام شود result = reverse_words("hello! world, how are you?") مقدار result به صورت زیر خواهد بود: olleh! dlrow, woh era uoy?
سوال ۱۲	یک لیست از اعداد صحیح به شما داده شده است. می‌خواهیم با استفاده از list comprehension (ساختن یک لیست جدید از روی لیست قبلی در ۱ خط)، لیستی بسازید که شرایط زیر را داشته باشد: ۱. تمام اعداد زوج از لیست اولیه را بردارد. ۲. اعداد زوج را به توان ۲ برساند. ۳. مربع اعداد زوج را تنها در صورتی نگه دارد که مضرب ۳ نباشند.



سوال ۱۳	تابعی بنویسید که با دریافت ۳ مجموعه، مجموعه دیگری را تحویل دهد که تنها شامل اعضای انحصاری هر یک از مجموعه‌ها است. به بیان دیگر، برای هر عضو مجموعه حاصل، تنها یکی از ۳ مجموعه ورودی اولیه وجود دارد که شامل آن عضو است.
سوال ۱۴	<p>عبارت منظمی بنویسید که با استفاده از آن، بتوان شماره تلفن‌ها را از یک متن استخراج کرد. دقت کنید که یک شماره تلفن صحیح، با موارد زیر سازگار است:</p> <p>۱. شماره تلفن ممکن است با یک کد کشور شروع شود. کد کشور (+) شروع می‌شود و پس از آن ارقام قرار می‌گیرند.</p> <p>۲. بخش اصلی شماره تلفن شامل ۳ گروه از ارقام است که توسط فاصله یا - از هم جدا می‌شوند.</p> <p>۳. هر گروه حاوی ۲ تا ۴ کاراکتر است.</p> <p>برای مثال این یک شماره تلفن است:</p> <p>+1-555-1234</p>
سوال ۱۵	<p>دیکشنری زیر در زبان پایتون را در نظر بگیرید.</p> <pre>sales_data = {     'product1': {'sales': [120, 150, 200, 180], 'discounts': [10, 15, 20, 18]},     'product2': {'sales': [80, 100, 120, 90], 'discounts': [8, 10, 12, 9]},     'product3': {'sales': [200, 220, 250, 230], 'discounts': [15, 18, 22, 20]} }</pre> <p>تابعی مانند aggregate_sales(sales_data) بنویسید که دیکشنری بالا را دریافت کند و مراحل زیر را روی آن انجام دهد.</p> <p>۱. مجموع فروش (sales) و مجموع تخفیف‌ها (discounts) را برای هر محصول (product) محاسبه کند.</p> <p>۲. محصولی را که بیشترین میزان فروش را داشته، شناسایی کند.</p> <p>۳. محصولی را که بیشترین تخفیف را داشته، شناسایی کند.</p> <p>تابع موردنظر باید خروجی به شکل زیر داشته باشد:</p> <pre>aggregated_data = {     'product1': {'total_sales': 650, 'total_discounts': 63,     'highest_total_sales': True, 'highest_avg_discount': False},     'product2': {'total_sales': 390, 'total_discounts': 39,     'highest_total_sales': False, 'highest_avg_discount': False},     'product3': {'total_sales': 900, 'total_discounts': 75,     'highest_total_sales': False, 'highest_avg_discount': True} }</pre>