



سوال ۱	واحد محاسبات و منطق (Arithmetic & Logic Unit) چه وظیفه‌ای را به عهده دارد و متعلق به کدام یک از اجزاء اصلی کامپیوتر است؟
	<p>واحد محاسبات و منطق یکی از اجزای اصلی پردازنده مرکزی (CPU) در کامپیوتر است و نقش کلیدی در انجام عملیات محاسباتی و منطقی دارد:</p> <p>۱. عملیات محاسبات ALU: مسئول انجام عملیات اصلی ریاضی مانند جمع، تفریق، ضرب، و تقسیم است.</p> <p>۲. عملیات منطقی: علاوه بر عملیات ریاضی، ALU همچنین عملیات منطقی مانند AND، OR، NOT و XOR را انجام می‌دهد.</p> <p>ALU به طور مستقیم در پردازش داده‌ها و اجرای دستورالعمل‌های برنامه‌های کامپیوتری نقش دارد و یکی از مهم‌ترین بخش‌های CPU به شمار می‌رود.</p>
سوال ۲	سه مورد از ویژگی‌های یک الگوریتم را نام ببرید و مختصر هر یک را توضیح دهید.
	<p>الگوریتم‌ها دارای ویژگی‌های متعددی هستند که از جمله مهم‌ترین آن‌ها عبارتند از:</p> <p>۱. مشخص بودن (Definiteness) هر یک از مراحل الگوریتم باید کاملاً مشخص و روشن باشد. دستورالعمل‌های الگوریتم باید به طور دقیق و بدون ابهام تعریف شده باشند تا هر بار که اجرا می‌شوند، نتایج یکسانی تولید کنند.</p> <p>۲. محدودیت (Finiteness) الگوریتم باید پس از تعداد محدودی از مراحل به پایان برسد. این بدان معناست که الگوریتم نباید به طور نامحدود ادامه یابد و باید در مدت زمان مشخصی به نتیجه یا خروجی برسد.</p> <p>۳. ورودی و خروجی (Input and Output) الگوریتم‌ها باید دارای ورودی باشند که بر اساس آن‌ها کار می‌کنند و خروجی تولید می‌کنند. ورودی‌ها داده‌هایی هستند که الگوریتم برای اجرای عملیات‌های خود از آن‌ها استفاده می‌کند و خروجی‌ها نتایج حاصل از اجرای الگوریتم هستند.</p>
سوال ۳	<p>پس از اجرای خط زیر، مقدار result چه خواهد بود؟ راه حل خود را مختصر توضیح دهید.</p> <p><code>result = 5 > 2 and 4 ** 2 != 16 or 6 % 3 + 9 % 4 > 3</code></p> <p>(راهنمایی: عملگرهای منطقی اولویت کمتری نسبت به بقیه عملگرها دارند).</p>
	<pre>5>2==>True 4**2 != 16==>False 6%3==>0 9%4==>1 0+1 > 3==> False True and False or False==>False or False==> False result = False</pre>
سوال ۴	<p>خروجی تکه کد زیر در زبان پایتون چیست؟ راه حل خود را توضیح دهید.</p> <pre>original = 40 result = 0 for num in range(4, 16, 4): print(result, end = " ") result = original - num</pre>



<pre>original = original + 2 print() print(original)</pre>																	
<p>خروجی به شکل زیر است:</p> <pre>0 36 34 46</pre> <p>کد را trace کنید.</p>																	
<p>عدد 431A1 در مبنای ۱۱ نوشته شده است. نمایش آن را در مبنای ۸ به دست آورید.</p>	<p>سوال ۵</p>																
<p>اول تبدیل به مبنای ۱۰ می‌کنیم. می‌شود ۶۲۷۸۹. سپس به مبنای ۸ می‌بریم می‌شود ۱۷۲۵۰۵.</p>																	
<p>یک محقق حوزه موسیقی تصمیم گرفته است که نُت‌ها را با استفاده از فرکانس‌هایشان نمایش دهد. او چند ملودی ساده طراحی کرده و آن‌ها را به شکل زیر در یک متغیر رشته‌ای در زبان پایتون ذخیره کرده است. یک مثال از ملودی‌های او به شکل زیر است:</p> <p>melody = '293.66-2-349.23-5-392.00-3-440.00-1-440.00-8-261.63-2'</p> <p>در هر رشته مشابه آنچه در بالا نوشته شده، ابتدا فرکانس متناظر یک نت و سپس مدت زمان نواختن آن نوشته می‌شود. در این مثال باید ابتدا نت D4 به مدت ۲ ثانیه نواخته شود، سپس نت F4 به مدت ۵ ثانیه و همین‌طور الی آخر. این محقق برای اینکه دیگران هم بتوانند ملودی‌اش را بنوازند، فرکانس‌های مورد استفاده را در جدول زیر آورده است.</p> <table border="1"> <thead> <tr> <th>فرکانس</th> <th>نت</th> </tr> </thead> <tbody> <tr> <td>261.63</td> <td>C4</td> </tr> <tr> <td>293.66</td> <td>D4</td> </tr> <tr> <td>329.63</td> <td>E4</td> </tr> <tr> <td>349.23</td> <td>F4</td> </tr> <tr> <td>392.00</td> <td>G4</td> </tr> <tr> <td>440.00</td> <td>A4</td> </tr> <tr> <td>493.88</td> <td>B4</td> </tr> </tbody> </table> <p>تابعی به نام decoder بنویسید که با استفاده از داده‌ساختار dictionary، هر رشته مانند melody را به گونه‌ای تغییر دهد که به جای فرکانس‌ها، نت متناظر آن‌ها نوشته شود. برای مثال:</p> <p>decoder(melody)</p> <p>مقدار رشته‌ای</p> <p>'D4-2-F4-5-G4-3-A4-1-A4-8-C4-2'</p> <p>را تحویل دهد.</p>	فرکانس	نت	261.63	C4	293.66	D4	329.63	E4	349.23	F4	392.00	G4	440.00	A4	493.88	B4	<p>سوال ۶</p>
فرکانس	نت																
261.63	C4																
293.66	D4																
329.63	E4																
349.23	F4																
392.00	G4																
440.00	A4																
493.88	B4																
<pre>def decoder(melody): freq2note = {'261.63': 'C4', '293.66': 'D4', '329.63': 'E4', '349.23': 'F4', '392.00': 'G4', '440.00': 'A4', '493.88': 'B4'} parts = melody.split('-') decoded_melody = '' for idx, part in enumerate(parts): if idx % 2 == 0: decoded_melody += freq2note[part] else:</pre>																	



<pre> decoded_melody += part if idx != len(parts)-1: decoded_melody += '-' return decoded_melody </pre>	
<p>سوال ۷</p> <p>می‌خواهیم تابعی را پیاده‌سازی کنیم که مقادیر پرت را از یک لیست حاوی اعداد حذف می‌کند و لیست جدید را تحویل می‌دهد. تعریف ما از مقادیر پرت به شرح زیر است: اگر لیست داده‌شده حاوی اعداد</p> <p>15, 17, 19, 23, 150, 14, 10, 80</p> <p>باشد، عدد ۱۵۰ را پرت می‌نامیم. زیرا بیش از ۲ برابر انحراف معیار، از میانگین فاصله دارد.</p> <p>یادآوری: انحراف معیار یک نمونه شامل اعداد، از طریق رابطه زیر به دست می‌آید:</p> $S = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$ <p>که در آن n تعداد اعداد، \bar{x} میانگین اعداد و x_i عدد i-ام است. در مثال فوق، انحراف معیار برابر است با:</p> $\bar{x} = \frac{15 + 17 + 19 + 23 + 150 + 14 + 10 + 80}{8} = 41$ $S = \sqrt{\frac{(15 - 41)^2 + (17 - 41)^2 + \dots + (80 - 41)^2}{7}} = 49/50036$ $150 - 41 = 109 > 2 * 49/50036$	
<pre> def outlier_detector(lst): mean = sum(lst)/len(lst) std = (sum([(num-mean)**2 for num in lst])/(len(lst)-1))**(1/2) new_lst = [num for num in lst if (num-mean)< 2*std] return new_lst </pre>	
<p>سوال ۸</p> <p>تابعی بنویسید که با دریافت یک عدد صحیح، مقسوم‌علیه‌های آن را چاپ کند.</p>	
<pre> def print_divisors(num): for i in range(1, num+1): if num % i == 0: print(i) </pre>	
<p>سوال ۹</p> <p>یک کد رنگی در مبنای ۱۶ (hexadecimal) به ۲ شکل قابل نوشتن است:</p> <p>۱. فرم کوتاه: مثلاً #A4C. یک علامت هشتگ، پس از آن یک عدد ۳ رقمی در مبنای ۱۶ که شامل ارقام ۰-۹ و حروف A-F است.</p> <p>۲. فرم بلند: مثلاً #B481AF. یک علامت هشتگ، پس از آن یک عدد ۶ رقمی در مبنای ۱۶ شامل ارقام ۰-۹ و حروف A-F است.</p> <p>عبارت منظمی (regular expression) بنویسید که بتوان با استفاده از آن، یک کد رنگی در مبنای ۱۶ را تشخیص داد.</p>	
<p>جواب‌های مختلفی قابل قبول است. یک نمونه را آورده‌ایم.</p> <pre> import re </pre>	



<pre>pattern = r"#[0-9A-Fa-f]{3}([0-9A-Fa-f]{3})?" test_string = "#A4C" match = re.match(pattern, test_string) if match: print("شد یافت مطابقت", match.group()) else: print("نشد یافت مطابقت")</pre>	
<p>عبارت منظمی بنویسید که بتوان به کمک آن، پلاک یک ماشین با الگوی زیر را در متن تشخیص داد.</p> <p>RJ14CV0002</p> <p>شرح الگو: دو حرف بزرگ - دو رقم - دو حرف بزرگ - ۴ رقم</p>	<p>سوال ۱۰</p>
<pre>[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}</pre>	
<p>تابعی بنویسید که تعداد دلخواهی عدد را دریافت کند و حاصل ضرب مقادیر نامنفی را تحویل دهد. توجه کنید که تعداد اعداد، ورودی تابع نیست.</p>	<p>سوال ۱۱</p>
<pre>def multiply_non_negative(*numbers): product = 1 for num in numbers: if num >= 0: product *= num return product</pre>	
<p>خروجی تکه کد زیر در زبان پایتون چیست؟ (مختصر شرح دهید).</p> <pre>data = {'set': {1, 2, 3}, 'list': [4, 5, 6], 'tuple': (7, 8, 9)} key_to_remove = 'list' data[key_to_remove].remove(5) data['tuple'] += (10,) data['set'].add(data[key_to_remove][0]) del data[key_to_remove][1] print(sorted(data['set']), data[key_to_remove], len(data))</pre>	<p>سوال ۱۲</p>
<p>خروجی به شکل زیر است:</p> <pre>[1, 2, 3, 4] [4] 3</pre>	
<p>تکه کد زیر در زبان پایتون را در نظر بگیرید.</p> <pre>for i in range(1, 6): for j in range(1, i + 1): if i % 2 == 0: print(f"{j} * {i} = {j * i}", end=' ')</pre>	<p>سوال ۱۳</p>



<pre>else: print(f"{i} ^ {j} = {i ** j}", end=' ') print()</pre> <p>این تکه کد را به نحوی بازنویسی کنید که به جای for در آن از while استفاده شود.</p>	
<pre>i = 1 while i < 6: j = 1 while j < i + 1: if i % 2 == 0: print(f"{j} * {i} = {j * i}", end=' ') else: print(f"{i} ^ {j} = {i ** j}", end=' ') j += 1 print() i += 1</pre> <p>توجه کنید که باید هیچ استفاده‌ای از for در جواب شما وجود نداشته باشد.</p>	
<p>فرض کنید متغیرهای زیر تعریف شده باشند.</p> <pre>names = ["Amanda", "Ali", "Sina", "Milad", "Maryam"] midterm_grades = [98, 95, 83, 87, 90] final_grades = [91, 94, 80, 84, 89]</pre> <p>با استفاده از comprehension و تابع zip، یک دیکشنری بسازید کلید آن نام شخص و مقدار متناظر کلید، یک دوتایی (tuple) از نمره‌ها باشد. به بیان دیگر، باید متغیری تعریف کنید که حاوی مقادیر زیر باشد.</p> <pre>grades = {'Amanda': (98, 91), 'Ali': (95, 94), 'Sina': (83, 80), 'Milad': (87, 84), 'Maryam': (90, 89)}</pre>	<p>سوال ۱۴</p>
<pre>grades = {name: (midterm, final) for name, midterm, final in zip(names, midterm_grades, final_grades)}</pre>	
<p>با استفاده از برنامه‌ریزی پویا (dynamic programming)، تابعی بنویسید که با دریافت یک رشته، تعداد زیررشته‌های مجزای آن را اعلام کند. یک زیررشته از رشته‌ی s، دنباله‌ای از کاراکترهای s است که با حذف تعدادی از کاراکترها بدون تغییر ترتیب بقیه کاراکترها به دست آمده باشد.</p> <p>مثال: رشته abc را در نظر بگیرید. زیررشته‌های آن عبارت‌اند از: "", a, b, c, ab, ac, bc (رشته خالی/تهی).</p>	<p>سوال ۱۵</p>
<pre>def count_distinct_subsequences(s): mod = 10 ** 9 + 7 # Initialize an array to store the counts of distinct subsequences dp = [0] * (len(s) + 1) # Empty subsequence is always present dp[0] = 1 # Dictionary to store the last index of each character last_occurrence = {}</pre>	



```
# Iterate through the string
for i in range(1, len(dp)):
    # If the current character has occurred before
    if s[i - 1] in last_occurrence:
        # Update the dp value using the previous occurrences
        dp[i] = (2 * dp[i - 1] % mod - dp[last_occurrence[s[i - 1]] - 1] + mod) % mod
    else:
        # If it's the first occurrence, double the previous count
        dp[i] = (2 * dp[i - 1]) % mod

    # Update the last occurrence index
    last_occurrence[s[i - 1]] = i

# Subtract 1 to exclude the empty subsequence
return (dp[-1] - 1) % mod

# Example usage
input_string = "abc"
result = count_distinct_subsequences(input_string)
print(f"The count of distinct subsequences in '{input_string}' is: {result}")
```