

Nama : I Putu Teddy Dharma Wijaya

UAS Praktikum Data Science

kerjakan 3 soal pada <https://datalemur.com/> dengan ketentuan sebagai berikut :

- tipe soal yang diambil bebas
- untuk tipe soal Statistics, Probability, dan Machine Learning anda dapat menggunakan penjelasan dengan formula (opsional) dan data sintetis untuk mendukung penjelasan dengan tools bebas.
- untuk tipe soal python dan sql anda dapat menggunakan code editor yang tersedia pada website untuk mendapatkan case dan data yang dimaksud oleh soal
- screenshot dan taruh pengerjaan anda pada dokumen ini dan simpan dalam bentuk PDF
- upload file pdf anda pada submission di e-learning

1. Factorial Formula [Microsoft Python Interview Question] – Python (Easy)

```
def factorial(n):  
    # function for factorial  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n - 1)  
  
# print the result  
print(factorial(5))  
# result : 120  
# expected : 120
```

Penjelasan : menggunakan konsep rekursif yang dimana akan memanggil fungsi dirinya sendiri hingga nilai n sama dengan 0

2. Fizz Buzz Sum [Google Python Interview Question] – Python (Easy)

```
def fizz_buzz_sum(target):  
    # problem link: https://datalemur.com/questions/python-fizz-buzz-sum  
    a, b = 3, 5  
    ans = 0  
    iter = 1  
    c, d = 3, 5  
    while(c < target or d < target):  
        c = a * iter  
        d = b * iter  
  
        if c < 10:  
            ans += c  
        if d < 10:  
            ans += d  
        iter += 1  
  
    return ans  
  
# expected : 23  
print(fizz_buzz_sum(10))
```

Penjelasan : menggunakan perulangan while untuk melakukan perkalian antara variabel c dengan variabel iter begitupun dengan variabel d. Jika salah satu kondisi masih bernilai true (< 10) maka lakukan terus perulangan jika kedua kondisi sudah false maka hentikan perulangan. Variabel ans akan menyimpan penjumlahan yang memenuhi ($c, d * \text{iter} < 10$)

3. Missing Integer [Amazon Python Interview Question] – (Easy)

```
import numpy as np
# probleme : https://datalemur.com/questions/python-missing-integer
def missing_int(input: list[int]) -> int:
    max_value = input[np.argmax(input)]
    missing_value = 0
    for i in range(max_value):
        if i not in input:
            missing_value = i
    return missing_value

# expected : 2
print(missing_int([0,1,3]))
```

Penjelasan : cari terlebih dahulu nilai max dari sebuah list, kemudian lakukan perulangan dengan range sama dengan max, selama iterasi lakukan check apakah nilai dari iterasi (i) ada di dalam list atau tidak jika tidak ada maka masukan ke variabel missing value dan jika ada maka lanjutkan iterasi

4. Factorial Trailing Zeroes [Microsoft Python Interview Question] - Medium

```
def trailing_zeroes(n):

    factorial = 1

    # search for factorial value
    for i in range(n):
        factorial = factorial * n
        n -= 1

    # convert to string
    factorial = str(factorial)

    # counting zero char
    count_zero = 0
    for char_i in factorial:
        if char_i == "0":
            count_zero += 1

    return count_zero

# 5! = 120 and 120 has exactly 1 trailing zero or 10! = 3628800 return 2, because there are two trailing zeroes.
print(trailing_zeroes(10))
```

Penjelasan : cari terlebih dahulu nilai dari $n!$ lalu hasil dari faktorial tersebut kemudian akan di convert menjadi sebuah string. Lakukan perulangan pada string tersebut dan kemudian check apakah berisi "0" dan jika berisi 0 maka count zero akan ditambah ($+=1$) dan jika tidak maka perulangan akan tetap dilanjutkan .

5. Largest Contiguous Subarray Sum [Akuna Capital Python Interview Question] - Medium

```
def max_subarray_sum(input):
    max = 0
    start_index = 0
    sum = 0
    for len_ in range(2, len(input)):
        while(start_index + len_ <= len(input)):
            array = input[start_index : start_index + len_]
            for element in array:
                sum += element

            if sum > max:
                max = sum
                start_index += 1
                sum = 0
            start_index = 0

    return max

# expected : 11, because 9 + 2
print(max_subarray_sum([-1, -3, 5, -4, 3, -6, 9, 2]))
```

Penjelasan : for akan bertindak sebagai perulangan yang akan menyatakan berapa banyak elemen yang akan diambil dari sebuah list, misal jika len_ bernilai 2 maka ambil 2 elemen yang ada di dalam list. Kemudian akan masuk ke perulangan while yang dimana jika start_index (index mulai ketika mengambil elemen + len_) <= len (input) maka akan masuk ke perulangan, dan kemudian start_index akan ditambahkan secara terus menerus hingga tidak memenuhi kondisi yang ada di perulangan while. Di dalam perulangan while akan di cek apakah sum > max jika benar maka nilai max akan digantikan dengan sum. Akhir dari fungsi ini adalah return nilai max.

6. Largest Prime Factor [Facebook Python Interview Question]

```
def largest_prime_factor(target):
    # problem links : https://datalemur.com/questions/python-largest-prime-factor
    max = 0
    for number in range(2, target):
        if target % number == 0:
            factor = 0
            for i in range(1, number + 1):
                if number % i == 0:
                    factor += 1
            if factor == 2:
                if max < number:
                    max = number
    return max

# expected : 7
print(largest_prime_factor(42))
```

Penjelasan : perulangan for pertama akan melakukan pengecekan apakah nilai dari target jika dimodulasikan dengan nilai number akan sama dengan 0. Jika iya kemudian nilai number akan dilakukan pengecekan apakah nilai dari variabel number merupakan bilangan prima atau tidak jika iya akan dicek kembali apakah nilai dari variabel number ini

lebih besar dari max jika iya maka nilai max akan diganti dengan nilai dari variabel number.

7. Merge Conflicts [Microsoft Python Interview Question]

```
def has_merge_conflict(pull_requests)-> bool:
    is_merged_conflict = False

    # first iteration
    for idx in range(1, len(pull_requests)):
        temp = pull_requests[idx]
        temp_2 = pull_requests[idx - 1]
        # checking merge
        for index, _ in enumerate(temp):
            if temp[index] > temp_2[index]:
                is_merged_conflict = True
                break

    return is_merged_conflict

print(f'result : {has_merge_conflict([[5, 10], [15, 40], [25, 50]])}')
print(f'EXPECTED : {True}')
print(f'result : {has_merge_conflict([[30, 40], [10, 20], [5, 8]])}')
print(f'EXPECTED : {False}')
```

Penjelasan : ambil nilai ke i dari masing masing dan temp_1 akan mengambil nilai dari elemen list [1...n] dan temp_2 akan mengambil nilai elemen [0....n -1]. Setelah itu akan masuk ke perulangan kedua yang dimana akan dicek jika apakah nilai dari salah satu temp_1 < dari temp_2 jika iya maka sudah pasti terjadi merge conflict dan is merged _conflict akan diset menjadi True, begitupun sebaliknya.

8. Smallest Multiple [Google Python Interview Question]

```
def smallest_multiple(target : list):
    # problem link : https://datalemur.com/questions/python-smallest-multiple
    smallest_multiple_result = 1
    # cara kerja sama seperti mencari KPK, jadi nilai[i] di modulus dengan nilai[i + idx] dengan idx = 1.. n - 1 dan nilai
    for i in range(len(target) - 1):
        for j in range(i + 1, len(target)):
            if target[j] % target[i] == 0 and target[i] != 1 and target[j] > target[i]:
                target[i] = 1
            elif target[i] % target[j] == 0 and target[j] != 1 and target[i] > target[j]:
                target[j] = 1
    # kalikan semua elemen yang ada di dalam list untuk mendapatkan hasil
    for number in target:
        smallest_multiple_result *= number
    return smallest_multiple_result

# expected : 60
print(smallest_multiple([1, 2, 3, 4, 5]))
```

Penjelasan : konsep dari penyelesaian ini sama dengan kpk yang dimana pada kode tersebut merupakan alur dalam pencarian nilai dari KPK (Kelipatan persekutuan terkecil).

9. Two Sum [Amazon Python Interview Question] – Medium

```
# problem link : https://datalemur.com/questions/python-two-sum

def two_sum(input: list[int], target: int) -> list[int]:
    answer = []
    for i in range(len(input)-1):
        for j in range(i + 1, len(input)):
            if input[j] + input[i] == target:
                answer.append([i, j])
    return answer[0]

# input = [1, 4, 6, 10], target = 11 Output: [0, 3]
print(two_sum([1, 4, 6, 10], 11))
```

Penjelasan : perulangan pertama akan melakukan iterasi bilangan pertama dan perulangan kedua akan melakukan perulangan selangkah lebih maju dari iterasi pertama ($i + 1$). Kemudian jika bilangan pertama + bilangan kedua = target maka index dari list akan disimpan di dalam list answer. Jika perulangan sudah selesai maka list answers akan dikembalikan

10. Precision/Recall Tradeoff [BCG Gamma Machine Learning Interview Question] – Easy

```
from sklearn.metrics import precision_score

pred = [1, 1, 0, 1, 0]
actual = [1, 1, 0, 1, 1]

# note : only for 2 class case
def recall(actual, pred):
    tp = 0
    fn = 0
    for idx, _ in enumerate(actual):
        if pred[idx] == 0:
            if actual[idx] == 1:
                fn += 1
        elif pred[idx] == 1:
            if actual[idx] == 1:
                tp += 1
    recall_result = tp / (tp + fn)
    return recall_result

def precision(actual, pred):
    tp = 0
    fn = 0
    for idx, _ in enumerate(actual):
        if pred[idx] == 1:
            if actual[idx] == 1:
                tp += 1
            else:
                fn += 1
    precision_result = tp / (tp + fn)
    return precision_result

print(f'manually : {precision(actual=actual, pred=pred)}')
print(f'scikit-learn library : {precision_score(actual, pred)}')
```

Penjelasan : pada soal ini, kita diperintahkan untuk membuat function yang dapat menghitung precision yang dimana formulanya adalah $TP / TP + FN$. Yang dimana jika hasil prediksi = 1 dan nilai actual = 1 maka TP akan bertambah 1 dan jika actual = 0 maka FN akan bertambah 1. Jika di run maka hasil function yang dibuat sendiri akan menghasilkan hasil yang sama dengan hasil precision dari library scikit learn

Code Link : <https://github.com/TeddyDharma/UAS-BISA-AI/tree/main>