

Digi-X Lightning Assessment for Interns?

August 21, 2019

1 Digi-X Lightning Assessment for Interns

1.0.1 Contact Details

Name: Bhanuka Manesha Samarasekara Vitharana Gamage

Email: bsam0002@student.monash.edu

Contact number: +60 17 689 4092

Starting date: After 15th November

Internship duration: 12 Weeks or more

Portfolio link (optional): <https://github.com/bhanukaManesha/Digi-X>

Section 1: You are tasked in building a new product, an online restaurant reservation platform - choose a front-end language/framework, back-end language/framework and a database management system. Tell us why you've chosen this stack:

- Front End Language: TypeScript
- Front End Framework: Angular
- Back End Language: TypeScript
- Back End Framework: NodeJS
- Database Management System: MySQL

The reason for using Angular as the front end framework would be its efficiency. Since Angular apps can do most of the work on the edge base devices, the server overload will be less.

The reason for choosing NodeJS is the consistency for both the back end and the front end codebase for the programmers. The backend server can be used to communicate with the database to validate the reservation without reloading the app and check for concurrent reservation and send requests to the customer in real time of all the reservations.

The reason to choose MySQL for the DBMS is the ease of use and it available for free. But if security is a major concern we can upgrade the DBMS to a better one such as MongoDB

Section 2: In the programming language of your choice, write the solution for the following questions:

1. Write a for loop that prints all the multiples of 3 and 5 up to 100:

3,5,6,9,10...100

```
[1]: for i in range(101):
      if i % 3 == 0 or i % 5 == 0:
          print(i,end = ', ')
```

0, 3, 5, 6, 9, 10, 12, 15, 18, 20, 21, 24, 25, 27, 30, 33, 35, 36, 39, 40, 42, 45, 48, 50, 51, 54, 55, 57, 60, 63, 65, 66, 69, 70, 72, 75, 78, 80, 81, 84, 85,

2. Sort the array[100,2,5,13,29,68,87,4,17,45,54] with as little lines of code as possible:

```
[2]: def countingSort(alist):
      max_value = max(alist)
      count = [0 for i in range(max_value+1)]
      # Counting the numbers in the array
      # O(M)
      for character in alist:
          count[character] += 1
      # O(M)
      alist = []
      # O(1)
      for i in range(len(count)):
          if count[i] != 0:
              # O(M)
              while count[i] > 0:
                  alist.append(i)
                  count[i]-=1
      # O(M)
      return alist
```

```
[3]: array = [100,2,5,13,29,68,87,4,17,45,54]
      sorted_array = countingSort(array)
      print(sorted_array)
```

[2, 4, 5, 13, 17, 29, 45, 54, 68, 87, 100]

This approach runs in $O(M)$ where M is the maximum number in the array. So in this case, both the space and time complexity would be $O(M)$. If the maximum number was a large number, or a float value, this sorting algorithm will be inefficient. The reason for this is it allocates M amount of space. And if all the numbers except for one number is small, this approach will create wasted space and will decrease the efficiency. In that case, it would be better to use a sorting algorithm such as Quick Sort.

3. Write a function that detects whether a string is a palindrome:

- a. 'tacocat' true
- b. 'racecar' true
- c. 'beanbag' false
- d. 'Hannah' true

```
[4]: string_list = ['tacocat', 'racecar', 'beanbag', 'Hannah']

for string in string_list:
    if string.lower() == string[::-1].lower():
        print('The string ' + string + ' is a palindrome')
    else:
        print('The string ' + string + ' is not a palindrome')
```

The string tacocat is a palindrome
The string racecar is a palindrome
The string beanbag is not a palindrome
The string Hannah is a palindrome

This algorithm will run in $O(N)$ where N is the length of a string, since the equal method compares each character in the string.