Assignment 2 Report

Question 1

- **Software/Hardware Used**

1. Microsoft Visual Studio Code

- **Requirements for each task in this section**

1. Write a script to extract the two columns and a calculated third column and save it in a new file.

2. Use that new file as an input to plot the graph. Write the second script for the graph plotting.

3. Write a third script to control the above scripts or do it individually and finally run a command for Gnuplot to plot the new graph.

1.1: Step-by-Step configuration/implementation

- To do this question, I first opened the csv file in read mode, created a dictreader object and then created 2 empty lists to store all the values from the 2 columns in the csv file. Then I iterated over each row in the csv file and appended the values to the 2 empty lists. Then I created a for loop so that every value in one of the lists divides by each value in the other list and multiplies it by 100 to get the PDR.

1.2: Screenshots

```
#opening the file in read mode
filename = open('Sample for task 1 - separate column view.csv', 'r')

#creating the dictreader object
file = csv.DictReader(filename)

#creating empty lists
TotalPkAtTxNode = []
PkreceivedAtRxNode = []

# iterating over each row and append
# values to the empty lists
for col in file:
    #column 5
    TotalPkAtTxNode.append(col['Packets'])
    #column 7
    PkreceivedAtRxNode.append(col['Rx Packets'])

#dividing column 7 and column 3 and multiplying it by 100
result_list = []
for i in range(0, len(TotalPkAtTxNode)):
    result_list.append(PkreceivedAtRxNode[i] / TotalPkAtTxNode[i] * 100)
```

Question 2

- **Software/Hardware Used**

1. Microsoft Visual Studio Code

- **Requirements for each task in this section**

1. Write a Program in python to extract data from any job search websites

2.1: Step-by-Step configuration/implementation

- For this question, I first made a function that gets each different web page to extract the information. The I made a function where I am passing soup values and using HTML code parameters. Then I use a for loop because it is a dynamic page, and I do not want to separate the code for each different and then extract the information from the webpage using the transform function. Then I saved my dictionary to a csv file called jobs.csv.

1.2: Screenshots

```python
# Function that can get to each different web page to extract the information
def extract(page):
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.64 Safari/537.36
    url = f'https://ie.indeed.com/jobs?q=python%20developer&l=dublin&start={page}'
    r = requests.get(url, headers)
    soup = BeautifulSoup(r.content, 'html.parser')
    return soup
```

```python
# Function where i am passing soup values and using HTML code parameters
def transform(soup):
    # Using the div variable which is representing the job container of all the searched jobs
    divs = soup.find_all('div', class_ = 'slider_container css-11g4k3a eu4oa1w0')
    for item in divs:
        title = item.find('a').text
        company = item.find('span', class_ = 'companyName').text
        # The salary is not given for all the job cards or search results
        # and if we add the salary tag it will give us errors so we need to make
        # try catch to handle errors
        try:
            salary = item.find('span', class_ = 'metadata salary-snippet-container').text
        except:
            salary = ''
        summary = item.find('div', {'class' : 'job-snippet'}).text.replace('\n', '')


        job = {
            'title': title,
            'company': company,
            'salary': salary,
            'summary': summary
        }
        joblist.append(job)
    return
```

```
joblist = []

# using a for loop beacuse its a dynamic page and dont want
# to separate code for each different page
# uses 4 pages 0, 10, 20 and 30
for i in range(0,40,10):
    print(f'Getting page, {i}')
    # extracting the information from the webpage using the function
    c = extract(0)
    transform(c)

# saving my dictionary to a CSV file
df = pd.DataFrame(joblist)
print(df.head())

# called jobs.csv
df.to_csv('jobs.csv')
```