1. Code a program in Python that replicates this illustration:

|  |
| --- |
| plain text: ABCXYZ  encrypted text: [0,1,2,23,24,25] |

1. Code a program in Python that replicates this illustration (this is a “+2” letter shift):

|  |
| --- |
| plain text: ABCXYZ  encrypted text: [2,3,4,25,0,1] |

1. Answer the questions following this code:

|  |
| --- |
| XML\_string = """  <kfc>  <nuggets pieces='6'>  <price>5.95</price>  </nuggets>  <nuggets pieces='24'>  <price>10</price>  </nuggets>  <chips>  <large>4.95</large>  <regular>2.95</regular>  </chips>  </kfc>  """  import xml.etree.ElementTree as ET  tree = ET.ElementTree(ET.fromstring(XML\_string))  print(tree.getroot().tag) #root element  print(tree.getroot().find('chips').find('regular').text)  for nuggets in tree.getroot().findall('nuggets'):  print(nuggets.get('pieces')) #attribute: pieces  for elements in nuggets:  print(elements.text) |

1. What is the output from this program in its current state? (hint – there are 6 lines of output)
2. The Python variable XML\_string contains an XML string. Identify in this XML at least one:
   1. Root element
   2. Element
   3. Text
   4. Attribute
3. Answer the questions following this code:

|  |
| --- |
| JSON\_string = """  {  "kfc": {  "nuggets":  [  {"pieces": 6, "price": 5.95},  {"pieces": 24, "price": 10}  ]  ,  "chips": {  "large": 4.95,  "regular": 2.95  }  }  }  """  import json  data = json.loads(JSON\_string)  print(data["kfc"]["chips"]["regular"])  print(list(data)[0]) #first key in dictionary - why??  for nuggies in data["kfc"]["nuggets"]:  if nuggies["pieces"] < 9:  print(nuggies["price"]) |

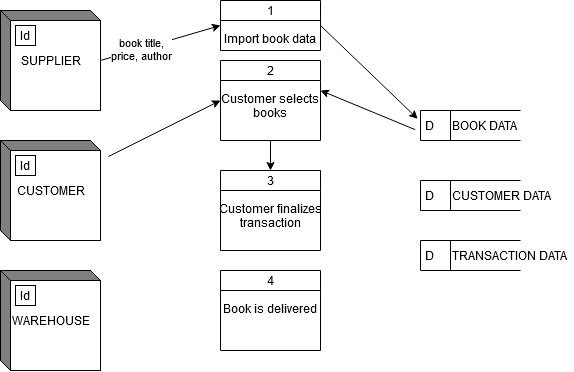
1. What is the output from this program in its current state? (hint – there are 3 lines of output)
2. The Python variable JSON\_string contains a JSON string. Identify in this JSON at least one:
   1. Key
   2. Value
   3. Array data type
   4. Key/value pairing separator (hint: it’s a type of punctuation mark)
3. Continue the following code snippet so that I can input another password string value, and the code should check what I input. If it matches the “**hashed\_password**” variable, the output should be “*Authenticated*.” Otherwise, the output should be “*Denied*.”

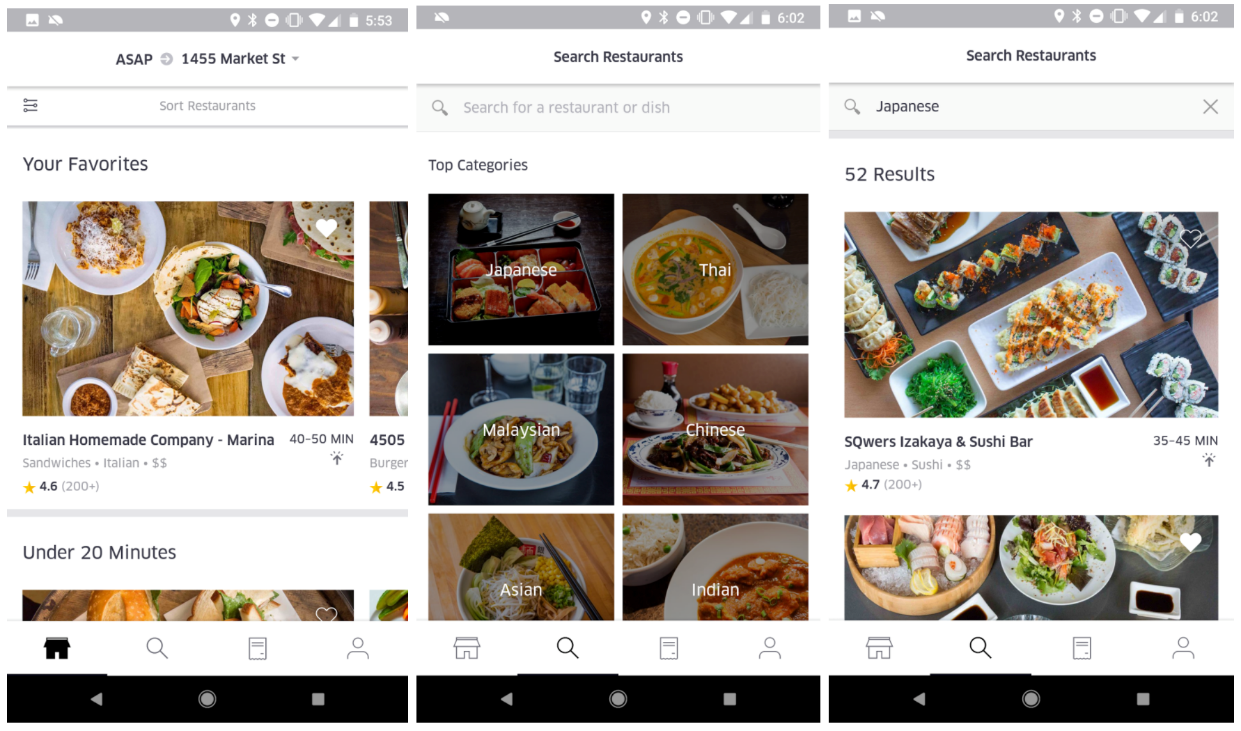
|  |
| --- |
| ######################## SALT #######################  #uuid 128bit sequence is (likely) going to be unique:  import uuid  salt = uuid.uuid4().hex.encode('utf-8')  #####################################################  plain\_text = input('enter password: ').encode('utf-8') #bytes  import hashlib  hashed\_password = hashlib.sha512(plain\_text+salt).hexdigest()  print('hashed password to record in database: ' + hashed\_password) |

1. Draw a feasible data flow diagram to illustrate the flow of data in this system:

|  |
| --- |
| An online bookshop system is needed to sell books online.  First, the online bookshop must *import* a catalogue of BOOK DATA from its SUPPLIER.  A CUSTOMER then *browses* the BOOK DATA via the web browser.  When the CUSTOMER *selects* a book to purchase, the details of the purchase are recorded as TRANSACTION DATA. Payment information is also requested from the customer.  The WAREHOUSE is responsible for accessing the TRANSACTION DATA and organising delivery.  Customers must save their CUSTOMER DATA to ensure correct delivery. |

Here is a start – feel free to modify:



1. **Uber Eats** app: imagine a future where Drones are used for deliveries. Sketch another 1 or 2 app screens where Uber Drone Pilot can control drone:

