|  |
| --- |
| **Program 01** |
| **Output** |
| Wallet Contents:  Credit Cards:  Visa  Discover  Master Card  IDs:  Drivers License  Student ID  Total cash is: $54  After changes, wallet contents are:  Credit Cards:  Visa  Master Card  IDs:  Drivers License  Student ID  Coupons:  Kohls shirt  Target toothpaste  Cub meat  Total cash is: $114  >>> |
| **Source Code** |
| #This program defines lists and dictionaries and uses them to  #manipulate a wallet of items. The wallet contains items of  #cash, coupons, ids, credit cards, and items can be added  #or removed. It then prints out the various items in their  #category type.  def main():  #Defining the wallet and the various items of the wallet  quantity= {'twenties': 2, 'tens': 1, 'fives': 0, 'ones': 4}  listOfCreditCards = ['Visa','Discover', 'Master Card']  listOfIDs = ['Drivers License', 'Student ID']  wallet = {'money':quantity, 'cards': listOfCreditCards, 'ids': listOfIDs}  #Initial print out of the contents of the wallet  print('Wallet Contents:')  print('\nCredit Cards:')  for i in range(0, len(listOfCreditCards)):  print(listOfCreditCards[i])  print('\nIDs:')  for i in range(0, len(listOfIDs)):  print(listOfIDs[i])  totalAmount = (quantity['twenties']\*20) + (quantity['tens']\*10) + (quantity['fives']\*5) + (quantity['ones']\*1)  print('\nTotal cash is: ${}'.format(totalAmount));  #Creates a coupon dictionary and adds items to it,  coupons = {}  coupons['Target'] = 'toothpaste'  coupons['Kohls'] = 'shirt'  coupons['Cub'] = 'meat'  wallet['coupons'] = coupons  quantity['twenties'] = (quantity['twenties'] + 3)  #Removes a card from the wallet  listOfCreditCards.remove('Discover')  #Prints the new contents of the wallet after additions and removals.\  print('\nAfter changes, wallet contents are:')  print('\nCredit Cards:')  for i in range(0, len(listOfCreditCards)):  print(listOfCreditCards[i])  print('\nIDs:')  for i in range(0, len(listOfIDs)):  print(listOfIDs[i])  print('\nCoupons:')  for key in coupons:  print (key, coupons[key])  totalAmount = (quantity['twenties']\*20) + (quantity['tens']\*10) + (quantity['fives']\*5) + (quantity['ones']\*1)  print('\nTotal cash is: ${}'.format(totalAmount));    if \_\_name\_\_ == "\_\_main\_\_": main() |

|  |
| --- |
| **Program 02** |
| **Output** |
| \*\*\*\*\*\*\*\*\*CUB Foods Inventory\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Items Price Quantity Value  Carrot $3 15 $45  Lettuce $2 0 $0  Onion $1.5 32 $48.0  Potato $4 6 $24  Your shopping bill is:  2 potatoes at $ 4 each - total $8  lettuce - out of stock.  1 onions at $ 1.5 each - total $1.5  The total bill is $ 9.5  >>> |
| **Source Code** |
| #This program utilizes three dictionaries. One for a shopping list,  #another for the grocers inventory, and a last one for the prices.  #The program takes these dictionaries and uses two methods. One  #to display inventory and prices and then another to show what  #the user wants to purchase and then calculates the bill and  #notifies the user for any items that are not in stock.  def main():  #predefined dictionaries  shoppingList = {'potatoes': 2, 'onions': 1, 'lettuce': 5}  inventory = {'potatoes': 6, 'lettuce': 0, 'onions': 32, 'carrots': 15}  prices = {'potatoes': 4, 'lettuce': 2,'onions': 1.5, 'carrots': 3}  #method calls  printInventory(inventory, prices)  computeBill(shoppingList, inventory, prices)    #print inventory method  def printInventory(stock, pricing):  print('\*\*\*\*\*\*\*\*\*CUB Foods Inventory\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('Items Price Quantity Value ')  print('Carrot ${} {} ${}'.format(pricing['carrots'],stock['carrots'], (pricing['carrots']\*stock['carrots'])))  print('Lettuce ${} {} ${}'.format(pricing['lettuce'],stock['lettuce'], (pricing['lettuce']\*stock['lettuce'])))  print('Onion ${} {} ${}'.format(pricing['onions'],stock['onions'], (pricing['onions']\*stock['onions'])))  print('Potato ${} {} ${}'.format(pricing['potatoes'],stock['potatoes'], (pricing['potatoes']\*stock['potatoes'])))  #compute the bill method  def computeBill(buyer, stock, pricing):  amount = 0    print('\nYour shopping bill is:\n')  #loop for going through the various keys of the dicitonary.  for key in buyer.keys():  if key in stock.keys() and stock[key] > 0:  print('{} {} at $ {} each - total ${}'.format(buyer[key],key , pricing[key], (buyer[key]\*pricing[key])))  amount += buyer[key]\*pricing[key]  else:  print('{} - out of stock.'.format(key))  print('\nThe total bill is $ {}'.format(amount))    if \_\_name\_\_ == "\_\_main\_\_": main() |

|  |
| --- |
| **Program 03** |
| **Output** |
|  |
| **Source Code** |
|  |

|  |
| --- |
| **Program 04** |
| **Output** |
|  |
| **Source Code** |
|  |

|  |
| --- |
| **Program 05** |
| **Output** |
|  |
| **Source Code** |
|  |