```
        reader = csv.DictReader(file)
        for row in reader:
            amount = float(row['Amount'])
            total_expenses += amount
            category = row['Category']
            if category in expense_categories:
                expense_categories[category] += amount
            else:
                expense_categories[category] = amount
    print("Expense Analysis:")
    print(f"Total Expenses: “,${total_exp})
```

DATA STRUCTURE – STACK

Data Structure: A data structure is a group of data which can be processed as a single unit. This group of data may be of similar or dissimilar data types. Data Structures are very useful while programming because they allow processing of the entire group of data as a single Unit.

Types of data structures:

Linear data structures: The elements are stored in a sequential order.

Example: Array, Stack, Queue.

Non-Linear data structures: The elements are not stored in sequential order.

Example: Graph, Tree, linked lists.

Stack: It is a data structure that allows adding and removing elements in a particular order. Every time an element is added, it goes on the top of the stack; the only element that can be removed is the element that was at the top of the stack.

Two Characteristics of Stacks: It is a LIFO (Last-In First-Out) data structure, The insertion and deletion happens at one end i.e. from the top of the stack.

Operations possible in the data structure: Major operations are Traversal, Insertion, Deletion and Searching.

Major operations on Stack:

1. PUSH: The addition of elements is known as PUSH operation. It is done using the TOP position.
2. POP: Removal of elements is known as POP operation. Removal of object is always done from TOP position.
3. PEEK: To show/ display the element placed at TOP position in the stack. Few applications of stack:

1. Expression evaluation
2. Backtracking (game playing, finding paths, exhaustive searching).
3. Memory management, run-time environment for nested language features.Stack implementation using List:

1. PUSH: The addition of elements is known as PUSH operation. It is done on the TOP position.

S= [‘element1’, ‘element2’, ‘element3’, ‘element4’]

S.append(‘newElement’) # pushing element in stack at the TOP

S= [‘element1’, ‘element2’, ‘element3’, ‘element4’, ‘newElement’] #List after insertion

2. POP: Removal of elements is known as POP operation. It is also done using the TOP position.

S= [‘element1’, ‘element2’, ‘element3’, ‘element4’, ‘newElement’]

S.pop() # removes element at top
S= [‘element1’, ‘element2’, ‘element3’, ‘element4’] #List after deletion
3. PEEK: To show/ display the element placed at TOP position in the stack.
return S[-1] # shows element at top but do not remove it from the stack.

Questions

1.	Consider a list named **Nums** which contains random integers. Write the following user defined functions in Python and perform the specified operations on a stack named **BigNums**. (i) **PushBig():** It checks every number from the list **Nums** and pushes all such numbers which have 5 or more digits into the stack, **BigNums.** (ii) **PopBig():** It pops the numbers from the stack, BigNums and displays them. The function should also display “**Stack Empty**” when there are no more numbers left in the stack. For example: If the list Nums contains the following data: **Nums**=[213, 10025, 167, 254923, 14, 1297653, 31498, 386, 92765] Then on execution of **PushBig(),**the stack **BigNums** should store: [10025, 254923, 1297653, 31498, 92765] And on execution of **PopBig(),** the following output should be displayed: 92765 31498 1297653 254923 10025 Stack Empty

ANS	def PushBig(Nums,BigNums):
	for N in Nums:
	if len(str(N))>=5:
	BigNums.append(N)
	def PopBig(BigNums):
	while BigNums:
	print(BigNums.pop())
	else:
	print(“Stack Empty”)

2.	A list, NList contains following record as list elements: [City, Country, distance from Delhi] Each of these records are nested together to form a nested list. Write the following user defined functions in Python to perform the specified operations on the stack named travel. (i) **Push_element(NList):** It takes the nested list as an argument and pushes a list object containing name of the city and country, which are not in India and distance is less than 3500 km from Delhi. (ii) **Pop_element():** It pops the objects from the stack and displays them. Also, the function should display “Stack Empty” when there are no elements in the stack. **For example: If the nested list contains the following data:** NList=[["New York", "U.S.A.", 11734], ["Naypyidaw", "Myanmar", 3219], ["Dubai", "UAE", 2194], ["London", "England", 6693],

	["Gangtok", "India", 1580], ["Columbo", "Sri Lanka", 3405]] **The stack should contain:** ['Naypyidaw', 'Myanmar'], ['Dubai', 'UAE'], ['Columbo', 'Sri Lanka'] **The output should be:** ['Columbo', 'Sri Lanka'] ['Dubai', 'UAE'] ['Naypyidaw', 'Myanmar'] Stack Empty

ANS	`travel = []`
	`def Push_element(NList):`
	`  for L in NList:`
	`  if L[1] != "India" and L[2]<3500:`
	`  travel.append([L[0],L[1]])`
	
	`def Pop_element():`
	`  while len(travel):`
	`  print(travel.pop())`
	`  else:`
	`  print("Stack Empty")`

3.	**(a) A list contains the following record of customer:** [Customer_name, Room Type] Write the following user-defined functions to perform given operations on the stack named ' Hotel': i) Push_Cust () - To Push customers names of those customers who are staying in Delux' Room Type. ii) Pop_Cust ()- To Pop the names of customers from the stack and display them. Also, display "Underflow" when there are no customers in the stack. For example: **If the lists with customer details are as follows:** ["siddarth", "Delux"] ["Rahul", "Standard"] ["Jerry", "Delux"] **The stack should contain** Jerry Siddharth **The output should be:** Jerry Siddharth Underflow b) Write a function in Python, Push (Vehicle) where, Vehicle is a dictionary containing details of vehicles - {Car_Name: Maker}. The function should push the name of car manufactured by "TATA' (including all the possible cases like Tata, TaTa, etc.) to the stack. For example: If the dictionary contains the following data: Vehicle={"Santro" : "Hyundai", "Nexon": "TATA", "Safari" : "Tata"} The stack should contain Safari Nexon

ANS

a) customer=[["Siddarth", "Delux"], ["Rahul", "Standard"], ["Jerry", "Delux"]]

```
hotel=[]
def push_cust():
  for i in customer:
    if i[1]=='Delux':
      hotel.append(i[0])
  return hotel

def pop_cust():
  if hotel==[]:
    return "Underflow"
  else:
    return hotel.pop()

push_cust()
while True:
  if hotel==[]:
    print(pop_cust())
    break
  else:
   print(pop_cust())
```

b) Vehicle={"Santro" : "Hyundai", "Nexon": "TATA", "Safari" : "Tata"}

```
stk=[]
def push(vehicle):
  for i in vehicle:
    if vehicle[i].lower()=='tata':
       stk.append(i)
    return stk

push(Vehicle)
for i in range(-1,-len(stk)-1,-1):
  print(stk[i])
```

4.

A list contains following record of a customer:
[Customer_name, Phone_number, City]

Write the following user defined functions to perform given operations on the stack named '**status**':
(i) Push_element() - To Push an object containing name and Phone number of customers who live in Goa to the stack
(ii) Pop_element() - To Pop the objects from the stack and display them. Also, display "Stack Empty" when there are no elements in the stack.

For example:
If the lists of customer details are:

["Gurdas", "99999999999","Goa"]
["Julee", "8888888888","Mumbai"]
["Murugan","77777777777","Cochin"]
["Ashmit", "1010101010","Goa"]

	The stack should contain [“Ashmit”,”1010101010”] [“Gurdas”,”9999999999”] The output should be: [“Ashmit”,”1010101010”] [“Gurdas”,”9999999999”] Stack Empty

ANS	status=[]
	def Push_element(cust):
	if cust[2]=="Goa":
	L1=[cust[0],cust[1]]
	status.append(L1)
	
	def Pop_element ():
	num=len(status)
	while len(status)!=0:
	dele=status.pop()
	print(dele)
	num=num-1
	else:
	print("Stack Empty")

5.	Write a function in Python, Push(SItem) where , SItem is a dictionary containing the details of stationary items– {Sname:price}. The function should push the names of those items in the stack who have price greater than 75. Also display the count of elements pushed into the stack. For example: If the dictionary contains the following data: Ditem={"Pen":106,"Pencil":59,"Notebook":80,"Eraser":25} The stack should contain Notebook Pen The output should be: The count of elements in the stack is 2

ANS	stackItem=[]
	def Push(SItem):
	count=0
	for k in SItem:
	if (SItem[k]>=75):
	stackItem.append(k)
	count=count+1
	print("The count of elements in the stack is : ", count)

6.	Julie has created a dictionary containing names and marks as key value pairs of 6 students. Write a program, with separate user defined functions to perform the following operations: ● Push the keys (name of the student) of the dictionary into a stack, where the corresponding value (marks) is greater than 75.

	● Pop and display the content of the stack. For example: If the sample content of the dictionary is as follows: R={"OM":76, "JAI":45, "BOB":89, "ALI":65, "ANU":90, "TOM":82} The output from the program should be: TOM ANU BOB OM

ANS	R={"OM":76, "JAI":45, "BOB":89, "ALI":65, "ANU":90, "TOM":82}
	def PUSH(S,N):
	S.append(N)
	def POP(S):
	if S!=[]:
	return S.pop()
	else:
	return None
	ST=[]
	for k in R:
	if R[k]>=75:
	PUSH(ST,k)
	while True:
	if ST!=[]:
	print(POP(ST),end=" ")
	else:
	break

7.	Alam has a list containing 10 integers. You need to help him create a program with separate user defined functions to perform the following operations based on this list. ● Traverse the content of the list and push the even numbers into a stack. ● Pop and display the content of the stack. For Example: If the sample Content of the list is as follows: N=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38] Sample Output of the code should be: 38 22 98 56 34 12

ANS	N=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38]
	def PUSH(S,N):
	S.append(N)
	def POP(S):
	if S!=[]:
	return S.pop()
	else:
	return None
	ST=[]
	for k in N:
	if k%2==0:
	PUSH(ST,k)
	while True:
	if ST!=[]:
	print(POP(ST),end=" ")
	else:
	break