UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

1. English Grades for Five Students

Aim

To read the English marks of five students and assign grades based on the given conditions.

Algorithm

Input: A list of English marks of five students
Output: The associated grades of the students

Pre-condition: The marks entered should be between 0 and 100.

Step 1: Start
Step 2: Define a function to assign English grades to the students. The function accepts the list of marks as the argument.
Step 3: The function contains a loop that traverses through every mark in the list and assigns a grade based on the given conditions. The grade is printed for each mark.
Step 4: In the main function, the required marks are obtained as input from the user and stored in a list.
Step 5: The list is passed to the function and the function is called.
Step 6: End

Program

```
# Python program to create a list of marks of 5 students in English
and assign the appropriate grades
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
# Function definition of function to assign grades based on the marks
scored by the student
def Assign_grade(Lst):
  for i in Lst:
    marks=i
    if marks>=90:
      print(marks,"Grade A")
    elif marks>=80 and marks<90:
      print(marks,"Grade B")
    elif marks>=65 and marks<80:
      print(marks,"Grade C")
    elif marks>=45 and marks<65:
      print(marks,"Grade D")
    else:
      print(marks,"Grade F")

print("English marks of 5 students:")
l=[]
for i in range(5):
  l.append(float(input("Enter mark: ")))
# Creating a list of the marks and passing it in the function call
Assign_grade(l)
```

Output

Input:

```
98 76 59 85 42
```

Output:

```
English marks of 5 students:
Enter mark: 98
Enter mark: 76
Enter mark: 59
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
Enter mark: 85
Enter mark: 42
98.0 Grade A
76.0 Grade C
59.0 Grade D
85.0 Grade B
42.0 Grade F
```

Results / Inferences

Program for reading marks of five students, storing them in a list and assigning the grades as required is written and executed.

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

2. Maximum and Minimum in a list

Aim

To enter elements into a list and to find the elements with the maximum and minimum values.

Algorithm

Input: A list
Output: The maximum and minimum elements in the list

Pre-condition: The number must be an integer.

Step 1: Start
Step 2: Define a function to find the maximum. The function accepts the list as the argument.
Step 3: The ‘maximum’ function does the following.
- Store the first element in a variable called ‘max’.
- Check if any element is greater than max and if yes, store that element in max. Repeat till the entire list is checked.
- Return the value of max.

Step 4: Define a function to find the minimum. The function accepts the list as the argument.
Step 5: The ‘minimum’ function does the following.
- Store the first element in a variable called ‘min’.
- Check if any element is smaller than min and if yes, store that element in min. Repeat till the entire list is checked.
- Return the value of min.

Step 6: In the main function, the list is obtained as input from the user.

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

Step 7: Call the functions to find the maximum and minimum, and print the return values.
Step 8: End

Program

```
# Python program to find the maximum and minimum of a user-input list of elements.

l=[]
n=int(input("Enter size of list: "))
for i in range(n):
  l.append(int(input("Enter num: ")))
print("List:",l)

def maximum(Lst):
  max=Lst[0]
  for i in Lst:
    if i>max:
      max=i
  return max

def minimum(Lst):
  min=Lst[0]
  for i in Lst:
    if i<min:
      min=i
  return min

print("Maximum:",maximum(l))
print("Minimum:",minimum(l))
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

Output

Input:

```
[1,2,3,4,5]
```

Output:

```
Enter size of list: 5
Enter num: 1
Enter num: 2
Enter num: 3
Enter num: 4
Enter num: 5
List: [1, 2, 3, 4, 5]
Maximum: 5
Minimum: 1
```

Results / Inferences

Program for finding the maximum and minimum values of the elements in a list is written and executed.

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

3. Reverse list

Aim

To reverse the positions of elements in a list without using slicing.

Algorithm

Input: A list
Output: The reversed list

Step 1: Start
Step 2: Define a function to reverse the list. The function accepts the list as its argument.
Step 3: Within the function, a for loop runs from 0 to half the size of the list to interchange the elements in the first and last positions, second and second last positions, etc., and so on till the middle element is reached.
Step 4: Read a list as input from the user.
Step 5: Call the function to reverse the list.
Step 6: Print the list after the function call.
Step 7: End

Program

```
# Python program to reverse a list without using slicing.

Lst=[]
n=int(input("Enter size of list: "))
for i in range(n):
  Lst.append(int(input("Enter num: ")))
print("List:",Lst)
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
# Function definition – to reverse the list
def print_reverse(Lst):
  for i in range(n//2):
    Lst[i],Lst[-i-1]=Lst[-i-1],Lst[i]

print_reverse(Lst)
print("Reverse list:",Lst)
```

Output

Input:

[21,42,54,18]

Output:

```
Enter size of list: 4
Enter num: 21
Enter num: 42
Enter num: 54
Enter num: 18
List: [21, 42, 54, 18]
Reverse list: [18, 54, 42, 21]
```

Results / Inferences

Program for reversing a list is written and executed.

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

4. Searching a list

Aim

To create a list of names and append an extra name to it; to find any required names using linear search and binary search.

Algorithm

Input: A list of 10 names
Output: Search results for a given name

Step 1: Start
Step 2: Read a list as input from the user. The list should contain 10 names.
Step 3: Read an extra name as input from the user and append it to the original list using the append() method.
Step 4: Read the name to search for as input from the user.
Step 5: Check if the name to search for is present in the name-list using the membership operator. If yes, use the index() method to find the position of the name in the list and print that the name is found at that position. If no, print that the name was not found.
Step 6: To search for a name using binary search, first, sort the name-list in ascending order and print the sorted list.

- Create a nested loop with two loop variables, say, i and j, ranging from 0 to the length of the list; i for the outer loop, j for the inner loop.
- If the j index-valued element is greater than the i index-valued element, then interchange the i^{th} and j^{th} elements.

Step 7: To perform binary search, proceed as such.

- The sorted list is divided into two halves to limit the search space.

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

- A while loop is run for as long as the element is not found or until the entire list has been searched.
- Check if the middle element is equal to the name to be searched for.
- If yes, take note of the position and exit the loop.
- If no, check if the middle element is lexicographically after or before the required name.
- If it is greater, then the search space is now confined to the first half of the list and the new middle element is now compared with the required name.
- If it is smaller, then the search space is now confined to the second half of the list and the new middle element is now compared with the required name.
- The loop continues till the element is found or the entire list has been searched.

Step 8: If the name is found, print the position that it was found at. Else, print that it was not found.
Step 9: End

Program

```
# Python program to create a list of names and find any given names using linear search and binary search.
#to create a list of names of 10 students
names=[]
n=10
for i in range(n):
  names.append(input("Enter name: "))
print(names)

#to append one more student
names.append(input("Enter extra name: "))
#to print the list
print(names)
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
#to search for a student: linear search
search=input("Enter name to search for: ")
if search in names:
  print("Found at position",names.index(search)+1)
else:
  print("Not found")
#to sort the names in alphabetical order
for i in range(len(names)):
  for j in range(len(names)):
    if names[j]>names[i]:
      names[i],names[j]=names[j],names[i]
print("Sorted:",names)
#to perform binary search
s=input("Enter name to search for: ")
beg,last=0,len(names)
flag=0
while(beg<last):
  mid=(beg+last)//2
  if names[mid]==s:
    pos=mid+1
    flag=1
    break
  elif names[mid]>s:
    last=mid
  else:
    beg=mid+1
if flag==1:
  print("Found at position",pos)
else:
  print("Not found")
```

Output

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
Enter name: Anne
Enter name: Diana
Enter name: Jane
Enter name: Priscilla
Enter name: Josie
Enter name: Ruby
Enter name: Millicent
Enter name: Dora
Enter name: Rilla
Enter name: Joyce
['Anne', 'Diana', 'Jane', 'Priscilla', 'Josie', 'Ruby',
'Millicent', 'Dora', 'Rilla', 'Joyce']
Enter extra name: Leslie
['Anne', 'Diana', 'Jane', 'Priscilla', 'Josie', 'Ruby',
'Millicent', 'Dora', 'Rilla', 'Joyce', 'Leslie']
Enter name to search for: Joyce
Found at position 10
Sorted: ['Anne', 'Diana', 'Dora', 'Jane', 'Josie',
'Joyce', 'Leslie', 'Millicent', 'Priscilla', 'Rilla',
'Ruby']
Enter name to search for: Leslie
Found at position 7
```

Results / Inferences

Program for searching for a name in a user-input list of names is written and executed.

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

5. Analysing a list

Aim

To create and analyse lists containing the names of students writing JEE and NEET.

Algorithm

Input: Lists of students writing JEE and NEET
Output: The required analysis of the lists

Step 1: Start
Step 2: Define a function to accept lists as input from the user. The function accepts an empty list and the number of participants (size of list) as arguments. It appends new elements and returns the final list.
Step 3: Two empty lists for JEE and NEET are created with their sizes being user input. The function to create the lists is called twice.
Step 4: Using the lambda function,

- names present in both the JEE and NEET lists are copied into another list.
- names present in the JEE list, but not the NEET list are copied into another list.
- names present in the NEET list, but not the JEE list are copied into another list.
- names present in either of the lists are added to another list to get the union of the two.

Step 5: The lists are printed.
Step 6: End

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

Program

```
# Python program to analyse two lists containing the names of
students writing JEE and NEET.

def createlist(l,s):
  for i in range(s):
    l.append(input("Enter name: "))
  return l

jee,neet=[],[]

n=int(input("Students passing JEE: "))
print("JEE list:",createlist(jee,n))

m=int(input("Students passing NEET: "))
print("NEET result:",createlist(neet,m))

a=list(filter(lambda x: x in jee,neet))
b=list(filter(lambda x: x not in neet,jee))
c=list(filter(lambda x: x not in jee, neet))
d=list(set(jee+neet))

print(a)
print(b)
print(c)
print(d)
```

Output

```
Students passing JEE: 3
Enter name: Bhargavi
Enter name: Kayal
Enter name: Anupama
JEE list: ['Bhargavi', 'Kayal', 'Anupama']
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
Students passing NEET: 4
Enter name: Ponni
Enter name: Bhavna
Enter name: Anupama
Enter name: Kayal
NEET result: ['Ponni', 'Bhavna', 'Anupama', 'Kayal']
['Anupama', 'Kayal']
['Bhargavi']
['Ponni', 'Bhavna']
['Bhavna', 'Bhargavi', 'Kayal', 'Anupama', 'Ponni']
```

Results / Inferences

Program for analysing two lists as required is written and executed.

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

6. List Operations

Aim

To perform the required operations on a list of students registered for a Python course.

Algorithm

Input: A list of names of students
Output: The results of the list operations performed on the list of students

Step 1: Start
Step 2: Read a list as input from the user.
Step 3: Read the name of the new student from the user and append it to the list using the append() method.
Step 4: Count the number of students who registered by finding the length of the list.
Step 5: Sort the list using the sort() method and print it.
Step 6: Print the elements with the index values 0 and -1 for the lexicographically first and last names respectively.
Step 7: Create an empty list and append the user-input new names of students joining the course late. Use the extend() method to attach this new list to the original list of students.
Step 8: Sort the list again and print it.
Step 9: Run a loop to traverse every name in the list and to read a register number for every student as input from the user and to store them in a list. Then, print the list of register numbers.
Step 10: To insert a new student at a given position, read the name, register no. and required position as input from the user. Then use the insert() method to insert the details into the names and register no. lists.

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

Step 11: Read the name of a student to search for. Check if the name exists in the list using the membership operator and if yes, print that it was found.
Step 12: Read the name of the student to be removed from the course. Use the remove() method to remove the name and the corresponding register no. from their respective lists.
Step 13: Display the details of the students writing the python exam by printing the register numbers and names of the students in the course.
Step 14: End

Program

```
# Python program to perform the required operations on a list of
students registered for a Python course

S=[]
n=int(input("No. of students registered: "))
#to input the names
for i in range(n):
  S.append(input("Enter name: "))
print(S)

#to enter a new student
S.append(input("Enter new name: "))

#to count the number of students registered
print("No. registered:",len(S))

#to sort the list
S.sort()
print(S)
#to find the first and last student in lexicographical order
print("First:",S[0])
print("Last:",S[-1])
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
#to modify the name list
N=[]
m=int(input("No. of new students registered: "))
  #getting the new names
for i in range(m):
  N.append(input("Enter name: "))
  #adding to the original list
S.extend(N)

#to sort the list again
S.sort()
print(S)

#to assign register numbers to all the students
rno=[]
print("Register numbers: ")
for i in S:
  print(i,end=':')
  rno.append(int(input()))
print(rno)

#to insert a new student at a given position
new=input("New name: ")
newrn=int(input("New register no.: "))
pos=int(input("Positon: "))
S.insert(pos-1,new)
rno.insert(pos-1,newrn)
print(rno)
print(S)

#to search for a student
search=input("Enter name to search for: ")
for i in S:
  if i==search:
    print("Found")
```

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
#to remove a student from the list
rem=input("Enter name to remove: ")
r=S.index(rem)
S.remove(rem)
rno.remove(rno[r])
print(S)

#to display the name-list of students for the Python exam
for i in range(len(S)):
  print(rno[i],S[i])
```

Output

```
No. of students registered: 4
Enter name: Simha
Enter name: Garud
Enter name: Mohak
Enter name: Adi
['Simha', 'Garud', 'Mohak', 'Adi']
Enter new name: Sampati
No. registered: 5
['Adi', 'Garud', 'Mohak', 'Sampati', 'Simha']
First: Adi
Last: Simha
No. of new students registered: 2
Enter name: Rohini
Enter name: Tara
['Adi', 'Garud', 'Mohak', 'Rohini', 'Sampati', 'Simha',
'Tara']
Register numbers:
Adi:1
Garud:2
Mohak:3
Rohini:4
Sampati:5
```

Krithika Swaminathan
Date: 16/02/2021
S03018

Ex. No.: 8

```
Simha:6
Tara:7
[1, 2, 3, 4, 5, 6, 7]
New name: Jayant
New register no.: 8
Positon: 5
[1, 2, 3, 4, 8, 5, 6, 7]
['Adi', 'Garud', 'Mohak', 'Rohini', 'Jayant', 'Sampati',
'Simha', 'Tara']
Enter name to search for: Mohak
Found
Enter name to remove: Garud
['Adi', 'Mohak', 'Rohini', 'Jayant', 'Sampati', 'Simha',
'Tara']
1 Adi
3 Mohak
4 Rohini
8 Jayant
5 Sampati
6 Simha
7 Tara
```

Results / Inferences

Program for performing the required list of operations on a list of students who registered for a course is written and executed.