

Name: Hinge Balasahab Narayan

Class: SE Comp-A

Roll No: 21C0047

Batch: B

Assignment-1

* Problem statement:-

Write a Python Program to Compute following operations on string:

- To display word with the largest length.
- To determine the frequency of the occurrence of particular character on the string.
- To check whether given string is a pallindrome or not.
- To display index of first appearance of the Substring.
- To count the occurrence of each word in a given String.

* Theory:-

i) strings

Strings in Python are surrounded by either single quotation mark or double quotation marks

e.g: 'Hello world' is the same as "Hello world"

To display a string we use `print()` function.

E.g:-
`a = "Hello world"`
`print(a)`

ii) Accessing a String value:

A single character is simply a string with a length of 1

• square brackets `[]` can be used to access elements of the string.

E.g:- `a = "Hello world"`

`print(a[1])`

Output:- e

iii) Looping through a string:-

Since strings are arrays, we can loop through the characters on a string, with a for loop.

E.g:-

`for x in "Apple":`

Output: A
P
P
l
e

iv) String length:-

To get the length of a string use the `len()` function.

Eg:-

```
a = " DSL assignment "  
print (len(a))
```

Output:- 14

v) Negative indexing:-

Use negative index to start the slice from the end of the string

Example:- `b = " DSL assignment "`
`print (b [-5:-2])`

Output:- nme

vi) Dictionary:-

Dictionaries are used to store data values on key: value pairs.

They are written with curly brackets.

Example:-

```
this dict = {  
    "brand": "Ford"  
    "Model": "Mustang"  
    "year": 1964  
}  
print (this dict)
```

output:

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
```

vii) Dictionary items:

Dictionary items are ordered, changeable and does not allow duplicate.

Dictionary items are presented on key value pairs, and can be referred to by using the key name.

Example:-

```
thisdict = {  
    "name": "Balasahab H",  
    "roll number": "21C0047",  
    "batch": "B".  
}  
print(thisdict["roll number"])
```

Output:

21C0047

viii) Python for Loops:

A loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string) with the loop we can execute a set of statements, once for each item in a list, tuple, set etc.

Example:-

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```

output: apple

banana

cherry

* Algorithm

a)

- 1) Read the list of words.
- 2) Increment the counter for each letter to count the no. of letters in each word.
- 3) Return the word with the highest no. of letters.

b):

- 1) Read the string and the character
- 2) Increment the counter whenever the character is found in the string.
- 3) Return this count after reaching the end of the string.

c):

- 1) Read the string.
- 2) Make the reverse string by concatenation of the letters of original string from last index to 0th index.
- 3) Compare the reverse string with original string if it is equal, then the given string is palindrome.

d):

- 1) Read the string and substring.
- 2) Read the first character of the substring with each character of the string and increment a counter.

- 3) If character is matched, check the subsequent if not, matched, continue step 2.
- 4) Repeat step 3 till you read end of ^{sub}string.
- 5) If all characters of substring match a part of a string,
return the counter.
- Else, return "The string substring does not lie in the string".

e):

- 1) Read the string.
- 2) Make a dictionary with the key-value pairs to be as word-occurrence.
- 3) Take the first word and compare it with the rest of the words. increment a counter if matched to count its occurrence.
- 4) Add the word & its occurrence in the dictionary
- 5) Repeat if a word already exists as a key in a dictionary, skip it.

Conclusion:- we have successfully completed the python program for to compute different operations on string.

Name: Hinge Balasaheb Narayan.

Class: SE Comp-A

Batch: 'B'

Roll No: 21C0047

Assignment - 2

* Problem Statement :-

Write a python Program to store first year percentage of students in array. Write function for sorting array of floating point number in ascending order using

a) Selection Sort.

b) Bubble sort and display top five scores.

Theory:-

Selection Sort:-

- 1) Selection Sort is a simple sorting algorithm. This sorting algorithm is an in-place comparison based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end. Initially, the sorted part is empty and the unsorted part is the entire list.
- 2) The smallest element is selected from the unsorted array and swapped with the leftmost element, and that element becomes a part of the sorted array. This process continues moving unsorted array boundary by one element to the right.
- 3) This algorithm is not suitable for large data sets or its average or worst case complexities are of $O(n^2)$, where n is no. of items.

* Bubble Sort:-

- 1) Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent element is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of $O(n^2)$ where n is the number of items.

Ex. :-

```
begin BubbleSort (list)
  for all elements of list
    if list[i] > list[i+1]
      swap (list[i], list[i+1])
    end if
  end for.
  return list
end Bubble sort
```


Algorithm:-

a) selection sort

- 1) set MIN to location 0
- 2) Search the minimum element in the list
- 3) Swap the value at location MIN
- 4) Increment MIN to point to next element.
- 5) Repeat until list is sorted.

b) Bubble sort

- 1) START
- 2) Repeat 3 for $i=0$ to N
- 3) Repeat For $J=0$ to $N-1$
- 4) IF $A[J] < A[J+1]$
 [END OF INNER LOOP]
 [END OF OUTER LOOP]
- 5) EXIT
 STOP

* Input:-

Enter the No. of students: 6

Enter the percentage of student 1: 36.6

Enter the percentage of student 2: 11.2

Enter the percentage of student 3: 99.9

Enter the percentage of student 4: 48.8

Enter the percentage of student 5: 56.30

Enter the percentage of student 6: 39.9

Output:-

-- -- -- SELECT A OPTION == -- -- --

1. SELECTION SORT

Ascending order

2. SELECTION SORT

Descending order

3. BUBBLE SORT to display top 5 score

4. EXIT

Enter your choice : 1

Selection sort

Sorted list in ascending order:

[11.2, 36.6, 39.9, 48.8, 56.30, 99.9]

Enter your choice : 2

Selection sort

Sorted list in descending order:

[99.9, 56.30, 48.8, 39.9, 36.6, 11.2]

Enter your Choice : 3

Bubble sort

Top 5 Scores [99.9, 56.30, 48.8, 39.9, 36.6, 11.2]

Enter your Choice : 4

THANK YOU!!

Conclusion :- we have successfully completed the Python program for sorting array of floating point number in ascending order using selection sort and bubble sort.

Name: Hinge Balasaheb Narayan

Class: SE COMP-A , Roll NO: 21C0047 Batch: 'B'

Assignment-3

Problem statement:-

write a python program to store first year percentage of students in array. write function for sorting array of floating point numbers in ascending order using quicksort and display top five scores.

Theory:-

String is a way of arranging items in a systematic manner. Quicksort is the widely used sorting algorithm that makes $n \log n$ comparisons in average case for using sorting an array of n elements.

- It is a faster and highly efficient sorting algorithm. This algorithm follows the divide and conquer approach. Divide and conquer is a technique of breaking down the algorithms into subproblems, then solving the subproblems and combining the results back together to solve the original problem.

- Divide: In Divide, first pick a pivot element. After that, position or rearrange the array into two sub-arrays such that each element in the left sub-arrays is less than or equal to the pivot element and each element in the right sub-array is larger than the

pivot element.

- Conquer:- Recursively sort two sub-arrays with quicksort
- Combine :- Combine the already sorted array

Algorithm :-

Partition (array, start, end)

1. pivot = a[start]
2. i = i + 1
3. for j start to end - 1, {
4. do if (a[j] < pivot) {
5. then i = i + 1
6. Swap a[i] with a[j]
7. }
8. Swap a[i + 1] with a[end]
9. return i + 1

quicksort (array, start, end)

1. if (start < end)
2. {
3. p = partition (a, start, end)
4. quicksort (a, start, p - 1)
5. quicksort (a, p + 1, end)
6. }
7. }

Input:-

Enter no. of students : 7

Enter percentage of student 1: 98.64

Enter percentage of student 2: 64.67

Enter percentage of student 3: 36.68

Enter percentage of student 4: 65.32

Enter percentage of student 5: 68.90

Enter percentage of student 6: 99.75

Enter percentage of student 7: 57.89

Output:-

- - - - SELECT A OPTION - - - -

1. Display Percentage of students.
2. quicksorted list to display top 5 scores.
3. Exit

Enter your Choise : 1

98.64

64.67

36.68

65.32

68.90

99.75

57.89

Enter your choice : 2

percentages of students quick sorted.

36.68

57.89

64.67

65.32

68.90

98.64

99.75

to display Top 5 scores choose (Y/N) : Y

Top 5 scores:

99.75

98.64

68.90

65.32

64.67.

Enter your choice : 3

THANK YOU!!!

Conclusion: We have Successfully Completed the python program for quick sort. to display the list in ascending order and to display top five scores.

Assignment-4

Name: Hinge Balasaheb Narayan

Class: SE Comp-A

Roll No: 210047

Aim:- queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. IF the operating system does not use priorities, then the jobs are processed in the order they enter the system. write C++ program for simulating job queue. write functions to add and delete job from queue.

Theory: queue is a linear data structure. It is considered as sequence of items. It supports FIFO (First in first out) Technique. It has three components:

- i) A certain items that contains element of queue
- ii) A pointer front that points the first item of the queue
- iii) A pointer rear that points the last item of the queue.

The insertion is performed from REAR end

The deletion is performed from front end

The insertion is also known as ENQUEUE in queue

- A queue can be implemented by two ways.

- Array or contiguous implementation
- linked list implementation

- Algorithm:-

- INIT (queue, front, Rear)

- IF (Rear = size)

- Print "queue is full"

- Else

- if (front = 0) and (Rear = 0) then
front = 1

- End if

- Rear = Rear + 1

- que[Rear] = Item

- End if

- Stop

- if (front = 0) then

- Print "queue is empty"

- IF (front = Rear)

- Rear = 0

- front = 0

- else

- front = front + 1

- End if

- End if

- stop.

* Input / output:

1. Insert element in queue
2. Delete Element in queue
3. Display element of queue
4. Exit.

Enter a choice: 1

Enter the job to be inserted: 2

Enter a choice: 1

Enter the job to be inserted: 3

Enter a choice: 1

Enter the job to be inserted: 4

Enter a choice: 3

2 3 4

Enter a choice: 2

Deleted Element: 2

Enter choice: 3

3 4

Enter a choice: 4

Thank You!!

* ~~Conclusion:-~~ We have successfully implement the operation to add and delete job from queue