

Started on	Tuesday, 22 July 2025, 1:23 PM
State	Finished
Completed on	Tuesday, 22 July 2025, 9:05 PM
Time taken	7 hours 41 mins
Overdue	5 hours 41 mins
Grade	100.00 out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Create a python program to find the longest palindromic substring using Brute force method in a given string.

For example:

Input	Result
mojologiccigolmojo	logiccigol

Answer: (penalty regime: 0 %)

Reset answer

```

1 def printSubStr(str, low, high):
2
3     for i in range(low, high + 1):
4         print(str[i], end = "")
5
6 def longestPalindrome(str):
7     n=len(str)
8     max_len=0
9     start=0
10    for i in range(n):
11        for j in range(1,n):
12            s=str[i:j+1]
13            if s==s[::-1]:
14                cur=j-i+1
15                if cur>max_len:
16                    max_len=cur
17                    start=i
18    printSubStr(str, start, start + max_len - 1)
19
20 if __name__ == '__main__':
21
22     str = input()

```

	Input	Expected	Got	
✓	mojologiccigolmojo	logiccigol	logiccigol	✓
✓	sampleelpams	pleelp	pleelp	✓

Passed all tests! ✓



Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

LONGEST COMMON SUBSTRING PROBLEM

The longest common substring problem is the problem of finding the longest string (or strings) that is a substring (or are substrings) of two strings.

Answer: (penalty regime: 0 %)

```

1 def LCS(X, Y, m, n):
2
3     maxLength = 0
4     endingIndex = m
5     lookup = [[0 for x in range(n + 1)] for y in range(m + 1)]
6     for i in range(1, m + 1):
7         for j in range(1, n + 1):
8             if X[i - 1] == Y[j - 1]:
9                 lookup[i][j] = lookup[i - 1][j - 1] + 1
10            if lookup[i][j] > maxLength:
11                maxLength = lookup[i][j]
12                endingIndex = i
13     return X[endingIndex - maxLength: endingIndex]
14 if __name__ == '__main__':
15     X = input()
16     Y = input()
17     m = len(X)
18     n = len(Y)
19     print('The longest common substring is', LCS(X, Y, m, n))

```

	Input	Expected	Got	
✓	ABC BABA	The longest common substring is AB	The longest common substring is AB	✓
✓	abcdxyz xyzabcd	The longest common substring is abcd	The longest common substring is abcd	✓

Passed all tests! ✓

Answer

Marks for this submission: 20.00/20.00.

Question 3

Correct

Mark 20.00 out of 20.00

Create a python program to compute the edit distance between two given strings using iterative method.

For example:

Input	Result
kitten sitting	3

Answer: (penalty regime: 0 %)

```

1 def mind(x,y):
2     m=len(x)
3     n=len(y)
4     # dp=[[0]*(n+1) for _ in range(m+1)]
5     dp = [[0] * (n + 1) for _ in range(m + 1)]
6     # print(dp)
7     for i in range(m+1):
8         for j in range(n+1):
9             if i==0:
10                dp[i][j]=j
11            elif j==0:
12                dp[i][j]=i
13            elif x[i-1]==y[j-1]:
14                dp[i][j]=dp[i-1][j-1]
15            else:
16                dp[i][j]=min(dp[i-1][j-1],dp[i][j-1],dp[i-1][j])+1
17        return dp[m][n]
18 x=input()
19 y=input()
20 print(mind(x,y))

```

	Input	Expected	Got	
✓	kitten sitting	3	3	✓
✓	medium median	2	2	✓

Passed all tests! ✓



Marks for this submission: 20.00/20.00.

Question 4

Correct

Mark 20.00 out of 20.00

Create a python program to find the longest common subsequence using Memoization Implementation.

For example:

Input	Result
AGGTAB GXTXAYB	Length of LCS is 4

Answer: (penalty regime: 0 %)

```

1 def lcs(x,y,m,n,dp):
2     if m==0 or n==0:
3         return 0
4     if dp[m][n]!=-1:
5         return dp[m][n]
6     if x[m-1]==y[n-1]:
7         dp[m][n]=1+lcs(x,y,m-1,n-1,dp)
8         return dp[m][n]
9     dp[m][n]=max(lcs(x,y,m,n-1,dp),lcs(x,y,m-1,n,dp))
10    return dp[m][n]
11 x=input()
12 y=input()
13 dp=[[-1]*(len(y)+1) for i in range(len(x)+1)]
14 print("Length of LCS is",lcs(x,y,len(x),len(y),dp))

```

	Input	Expected	Got	
✓	AGGTAB GXTXAYB	Length of LCS is 4	Length of LCS is 4	✓
✓	SAMPLE SAEMSUNG	Length of LCS is 3	Length of LCS is 3	✓
✓	saveetha sabeetha	Length of LCS is 7	Length of LCS is 7	✓

Passed all tests! ✓



Marks for this submission: 20.00/20.00.

Question 5

Correct

Mark 20.00 out of 20.00

Write a python program to implement quick sort using the last element as pivot on the list of float values.

For example:

Test	Input	Result
quickSort(arr,0,n-1)	5 3.2 1.5 9.6 4.1 5.9	Sorted array is: 1.5 3.2 4.1 5.9 9.6

Answer: (penalty regime: 0 %)

```

1 def partition(arr, low, high):
2     pivot = arr[high]
3     i = low - 1
4     for j in range(low, high):
5         if arr[j] < pivot:
6             i += 1
7             arr[i], arr[j] = arr[j], arr[i]
8     arr[i + 1], arr[high] = arr[high], arr[i + 1]
9     return i + 1
10 def quickSort(arr, low, high):
11     if low < high:
12         pi = partition(arr, low, high)
13         quickSort(arr, low, pi - 1)
14         quickSort(arr, pi + 1, high)
15 n = int(input())
16 arr = []
17 for _ in range(n):
18     arr.append(float(input()))
19 quickSort(arr, 0, n - 1)
20 print("Sorted array is:")
21 for num in arr:
22     print(num)

```

	Test	Input	Expected	Got	
✓	quickSort(arr,0,n-1)	5 3.2 1.5 9.6 4.1 5.9	Sorted array is: 1.5 3.2 4.1 5.9 9.6	Sorted array is: 1.5 3.2 4.1 5.9 9.6	✓
✓	quickSort(arr,0,n-1)	6 2.3 50.4 9.8 7.6 3.4 1.5	Sorted array is: 1.5 2.3 3.4 7.6 9.8 50.4	Sorted array is: 1.5 2.3 3.4 7.6 9.8 50.4	✓
✓	quickSort(arr,0,n-1)	8 2.3 1.5 6.4 9.8 7.6 4.2 3.8 1.4	Sorted array is: 1.4 1.5 2.3 3.8 4.2 6.4 7.6 9.8	Sorted array is: 1.4 1.5 2.3 3.8 4.2 6.4 7.6 9.8	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.