

Started on Friday, 16 May 2025, 8:20 AM

State Finished

Completed on Friday, 16 May 2025, 8:42 AM

Time taken 21 mins 14 secs

Grade 100.00 out of 100.00

Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to implement sudoku solver using backtracking to find the the safe position in the grid.

Answer: (penalty regime: 0 %)

Reset answer

```
1 SIZE = 9
2 matrix = [
3     [6,5,0,8,7,3,0,9,0],
4     [0,0,3,2,5,0,0,0,8],
5     [9,8,0,1,0,4,3,5,7],
6     [1,0,5,0,0,0,0,0,0],
7     [4,0,0,0,0,0,0,0,2],
8     [0,0,0,0,0,0,5,0,3],
9     [5,7,8,3,0,1,0,2,6],
10    [2,0,0,0,4,8,9,0,0],
11    [0,9,0,6,2,5,0,8,1]]
12 def print_sudoku():
13     for i in matrix:
14         print (i)
15 def number_unassigned(row, col):
16     num_unassign = 0
17     for i in range(0,SIZE):
18         for j in range (0,SIZE):
19             if matrix[i][j] == 0:
20                 row = i
21                 col = j
22                 num_unassign = 1
```

	Test	Expected	Got	
✓	solve_sudoku()	[6, 5, 1, 8, 7, 3, 2, 9, 4] [7, 4, 3, 2, 5, 9, 1, 6, 8] [9, 8, 2, 1, 6, 4, 3, 5, 7] [1, 2, 5, 4, 3, 6, 8, 7, 9] [4, 3, 9, 5, 8, 7, 6, 1, 2] [8, 6, 7, 9, 1, 2, 5, 4, 3] [5, 7, 8, 3, 9, 1, 4, 2, 6] [2, 1, 6, 7, 4, 8, 9, 3, 5] [3, 9, 4, 6, 2, 5, 7, 8, 1]	[6, 5, 1, 8, 7, 3, 2, 9, 4] [7, 4, 3, 2, 5, 9, 1, 6, 8] [9, 8, 2, 1, 6, 4, 3, 5, 7] [1, 2, 5, 4, 3, 6, 8, 7, 9] [4, 3, 9, 5, 8, 7, 6, 1, 2] [8, 6, 7, 9, 1, 2, 5, 4, 3] [5, 7, 8, 3, 9, 1, 4, 2, 6] [2, 1, 6, 7, 4, 8, 9, 3, 5] [3, 9, 4, 6, 2, 5, 7, 8, 1]	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Create a python program to find the length of longest common subsequence using naive recursive method

For example:

Input	Result
AGGTAB GXTXAYB	Length of LCS is 4

Answer: (penalty regime: 0 %)

```

1 def longestcommon(x,y,m,n):
2     if m==0 or n==0:
3         return 0
4     elif x[m-1]==y[n-1]:
5         return 1+longestcommon(x,y,m-1,n-1)
6     else:
7         return max(longestcommon(x,y,m,n-1),longestcommon(x,y,m-1,n))
8
9 X = input()
10 Y = input()
11 print ("Length of LCS is ", longestcommon(X , Y, len(X), len(Y)) )

```

	Input	Expected	Got	
✓	AGGTAB GXTXAYB	Length of LCS is 4	Length of LCS is 4	✓
✓	saveetha engineering	Length of LCS is 2	Length of LCS is 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 3

Correct

Mark 20.00 out of 20.00

Create a Python program to find longest common substring or subword (LCW) of two strings using dynamic programming with bottom-up approach.

A string r is a substring or subword of a string s if r is contained within s . A string r is a common substring of s and t if r is a substring of both s and t . A string r is a longest common substring or subword (LCW) of s and t if there is no string that is longer than r and is a common substring of s and t . The problem is to find an LCW of two given strings.

For example:

Test	Input	Result
lcw(u, v)	bisect trisect	Longest Common Subword: isect

Answer: (penalty regime: 0 %)

Reset answer

```

1 def lcw(x,y):
2     m = len(x)
3     n = len(y)
4
5     dp=[[0]*(n+1) for _ 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                 dp[i][j]=1+dp[i-1][j-1]
10            else:
11                dp[i][j]=0
12
13     maxl=0
14     end_i=0
15     end_j=0
16
17     for i in range(1,len(x)+1):
18         for j in range(1,len(y)+1):
19             if dp[i][j]>maxl:
20                 maxl=dp[i][j]
21                 end_i=i
22                 end_j=j

```

	Test	Input	Expected	Got	
✓	lcw(u, v)	bisect trisect	Longest Common Subword: isect	Longest Common Subword: isect	✓
✓	lcw(u, v)	director conductor	Longest Common Subword: ctor	Longest Common Subword: ctor	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 4

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
5     dp = [[0] * (n + 1) for _ in range(m+1)]
6
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! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 5

Correct

Mark 20.00 out of 20.00

Create a python program to find the longest palindromic substring using optimal algorithm Expand around center.

For example:

Test	Input	Result
findLongestPalindromicSubstring(s)	samsunggnusgnusam	sunggnus

Answer: (penalty regime: 0 %)

Reset answer

```

1 def expand(s, low, high):
2     length = len(s)
3     while low >= 0 and high < length and s[low] == s[high]:
4         low = low - 1
5         high = high + 1
6
7     return s[low + 1:high]
8
9
10 def findLongestPalindromicSubstring(s):
11
12     if not s or not len(s):
13         return ''
14
15     start = 0
16     end = 0
17
18     for i in range(len(s)):
19         len1 = expand(s, i, i)
20         len2 = expand(s, i, i + 1)
21
22         if len(len1) > len(len2):

```

	Test	Input	Expected	Got	
✓	findLongestPalindromicSubstring(s)	samsunggnusgnusam	sunggnus	sunggnus	✓
✓	findLongestPalindromicSubstring(s)	welcomeindiaaidni	indiaaidni	indiaaidni	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.