CS1010E Mid-term Test (AY2022/2023, SEM1)

QN	Questions	Answer
	Evaluate the following expression without any pre-defined variable or packages imported	
1	`5-3+2*4-1` A. 9 B. 15 C. 12 D. -16 E. 8	A
2	`6-------6` A. 0 B. 12 C. 3 D. -3 E. Error	A
3	`6-True+False**0` A. 6 B. 4 C. 5 D. ZeroDivisionError E. TypeError	A
4	`8/4*2` A. 4.0 B. 4 C. 1.0 D. 1 E. 0.5	A
5	`'1234567'[4]` A. '5' B. '7' C. '6' D. '4' E. '1234'	A
6	`'1234567'[2:5][1:2][1:]` A. ''	A

	B. '4' C. '34' D. '3' E. Error	
7	('abc','abc','def','def','ghi','jkl')[4] A. 'ghi' B. ('ghi',) C. 'def' D. ('def',) E. 'b'	A
8	tuple('xyz')+tuple((3)) A. Error B. ('x', 'y', 'z', 3) C. ('xyz', 3) D. (('x', 'y', 'z'), 3) E. (['xyz'], 3)	A
9	[1,2,[3,4],5,6][[1,2,4][2]:[1,2,3,4,5][3]] A. [] B. Error C. [[3, 4], 5] D. [[3, 4]] E. [3, 4]	A
10	(lambda x,y,z:return x-y+z)(3,2,1) A. Error B. 4 C. 0 D. 2 E. None	A
11	(lambda x,y:y(y(x))+y(x))(17,lambda x:x//2) A. 12 B. 5 C. 4 D. Error E. 21	A
12	(lambda x: x((lambda x: x(lambda x: x))(x(x))))(lambda x: x)(lambda x:x+x)(3) A. 6 B. 3	A

	C. RecursionError D. A function E. SyntaxError	
	If the following is in a .py file, what is the output in console if you execute/run it?	

13	x = 0
	y = 0
	while x < 5:
	    x += 1
	    y += 2
	print(y)
	A. 10
	B. 5
	C. 15
	D. 20
	E. 0
		A
		
		(This is a FITB question.)

14	q = 15
	if q > 5:
	    if q < 7:
	        print('a')
	    elif q > 9:
	        print('b')
	    elif q == 15:
	        print('c')
	    else:
	        print('d')
	A. 'b'
	B. 'a'
	C. 'c'
	D. 'd'
	E. Print nothing
		A

15	def f1(x):
	    return 1+f3(x)
	def f2(x):
	    return 2+f4(x)
	def f3(x):
	    return 1+f1(x)
	print(f1(4))
		A

	A. RecursionError B. NameError C. Infinite loop D. 8 E. 6	

16	def f1(x):
	    return '1'+f2(x)
	def f2(x):
	    return f3(x)+'2'
	def f3(x):
	    return '3'+f4(x)
	def f4(x):
	    return '4'+x
	print(f1(0))
	A. Error
	B. '13402'
	C. '3042'
	D. '13042'
	E. '12340'
		A

17	x = ['a','b','c','d']
	def foo(l,f):
	    if not l:
	        return l
	    return foo(f(l[1:]),f)+[f(l[0])]
	print(foo(x,lambda x:x[::-1]))
	A. ['c','b','d','a']
	B. ['a','b','c','d']
	C. ['d','c','b','a']
	D. ['a','d','b','c']
	E. ['a','c','b','d']
		A

18	d = {0:2, 1:5, 2:1, 3:4, 4:7, 5:6, 6:3, 3:9}
	a = 0
	output = ''
	while a in d:
	    a = d[a]
	    output += str(a)
	print(output)
		A

	A. '215639' B. '0215639' C. '2156347' D. '02156347' E. Infinite loop F. Error	
19	lst1 = ['bc','de','ya','ab','bq','bd'] lst2 = [] for x in lst1: lst2.append(tuple(x)) d = dict(lst2) print(d['b']) A. 'd' B. 'a' C. Error D. 'bc' E. 'ab'	A
20	x = {'a','bc','de','a'} y = {'b','de','a','a','b'} print(x|y-x^y) A. {'de', 'a', 'bc'} B. {'bc'} C. {'a', 'de', 'b', 'bc'} D. {'a', 'de', 'b'} E. {}	A
	Fill in the blanks	

21	Given a string s, we want to remove all <u>consecutive</u> duplicated characters. For example:
	aabbbbcccddabdd -> abcdabd
	Some sample output:
	>>> remove_duplicate('abcdeea')
	'abcdea'
	>>> remove_duplicate('aaaabbbbaaaa')
	'aba'
	Fill in the blanks for the missing part in the code to complete the following function as mentioned above:
	def remove_duplicate(s):
		Blank1: 2
		Blank 2:
		s[0] == s[1]
		(or s[1]==s[0])
		Blank 3:
		S[0]
		
		correct code:
		def remove_duplicate(s):
		  if len(s) < (2):

	`if len(s) < (__BLANK_1__) :`
	`  return s`
	`if (__BLANK_2__) :`
	`  return remove_duplicate(s[1:])`
	`else:`
	`  return (__BLANK_3__) + remove_duplicate(s[1:])`
		`  return s`
		`if ( s[0] == s[1] ):`
		`  return remove_duplicate(s[1:])`
		`else:`
		`  return ( s[0] ) + remove_duplicate(s[1:])`

22	Remember our Assignment 3:
	Write a recursive version of **`binom_coeff_recur(n,k)`** to compute the binomial coefficient by using recursion **without** using any factorial functions or loops. **You must use recursion**. The binomial coefficient can be expressed in another form:
	$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k},$$
	for $\binom{n}{n} = \binom{n}{0} = 1$.
	
	Fill in the blanks for the missing part in the code to complete the following function as mentioned above. In order to make your code simpler, we use the function name as `'nCk'` instead of the long one `'binom_coeff_recur'`:
	`def nCk(n,k):`
	`  if (__BLANK_1__ ):`
	`  return 1`
	`  return (__BLANK_2__) + (__BLANK_3__)`
		Blank 1:
		`k == 0 or n == k`
		
		Blank 2 and Blank 3:
		`nCk(n-1,k)`
		`nCk(n-1,k-1)`
		
		Correct complete code:
		`def nCk(n,k):`
		`  if (k == 0 or n == k):`
		`  return 1`
		`  return ( nCk(n-1,k)) + ( nCk(n-1,k-1) )`

23	Given that the input `L` is a list of integers with `len(L)` > 1, what does the function `foo(L)` do?
	
	`def foo(L):`
	`  for i in range(len(L)-1):`
	`  for j in range(len(L)-i):`
	`  if L[j]>L[j+1]:`
	`  L[j],L[j+1] = L[j+1],L[j]`
	
	A. The function actually always crashes. It won't work
	B. Sort the input list L in ascending order
	C. Sort the input list L in descending order
		A

	D. Push the largest elements to the end of the list, but the list may or may not be fully sorted E. Push the largest elements to the beginning of the list, but the list may or may not be fully sorted	
24	If we open a file with the file mode '`r+`', it means: A. Opens a file for both reading and writing. The file pointer will be at the beginning of the file B. Opens a file for both reading. The file pointer will be at the beginning of the file C. Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing. D. Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, it creates a new file for reading and writing E. Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.	A
25	How many of the following data type(s) ***cannot*** be store in the keys of a Python dictionary • int • float • bool • string • list • dict • tuple • set A. 3 B. 0 C. 2 D. 1 E. 8	A