Instructions: This quiz is ... of laptops is forbidden. Answer all questions. Questions using the sign ♣ may have zero, o... correct answers; negative points may be attributed to *very bad* answers. Other questions ... correct answer.

Allotted time is 20 minutes. Return this question paper along with your answer sheet. **Do no... soil these sheets, since they will be machine-graded.**

Total marks: 20

Question 1 (1 mark) If the following output is obtained on using the function `get_name` ... must be the content of the function?

```
def get_name(input_name):
    # function definition goes here...

>>> name = get_name('Turing')
Turing
>>> print(name)
Turing
```

A `print (input_name)`

B `return input_name`

[✓ C] `print (input_name)`
   `return input_name`

D `return input_name`
   `print (input_name)`

E None of the above

```
answer = a
self.front = (0+1)%10 = 1
self.size = 5-1 = 4

enqueue
avail = (1+4)%10 = 5
self.data [avail] = f
self.size = 5

dd = [k, b, c, d, e, f, g, h, i, j]
self._data = [None]*20
walk = self._front
for k in range (self._size):
```

Question 7 (1 mark) What happens when the following snippet of code is executed:

```
def f(x):
  return (x**x) + (x^x)

print(f(3))
```

A 54 [✓ B] 27 C 30 D None E Python throws an error

Question 8 ♣ (2 marks) When the following snippet of code is executed, which of the tests **fail**?

2

```
def num_test():
    ''' Test file
    >>> (3) + (4)    #test1
    7
    >>> (3,1) + (3) #test2
    (3, 1, 3)
    >>> [3,1] + [3] #test3
    [3, 1, 3]
    >>> (3)*10  #test4
    30
    >>> [3]*10  #test5
    [30]
    '''

import doctest
doctest.testmod(verbose = True)
```

A test1 [✓ B] test2 C test3 D test4 [✓ E] test5

Question 9 (1 mark) The expression `int(99.6)` produces the output

y

A 99.0 B 100 C '99' [✓ D] 99 E SyntaxError

Question 10 ♣ (2 marks) Based on your understanding of the Python function `round()`, which of the following statements will return True?

[✓ A] `round(94.5)==94`
[✓ B] `round(94.50000001)==95`
[✓ C] `round(94.9,-2)==100`
[✓ D] `round(94.4)== 94`
E `round(94.5)==95`

J. JANU SAHANA

Instructions: This quiz is 'closed book' — not even *your* hand-written class notes are permitted. The use of laptops is forbidden. **Write your answers only INSIDE the rectangles. The gray boxes are for the instructor's use only.** Answer all questions. Allotted time is 20 minutes.

Total marks: 20

Question 1 (3 marks) Consider a program that uses a Monte Carlo method to estimate π. For this program, the estimate of π from 10,000 iterations is guaranteed to be closer to the real value of π than after 9,000 iterations.

□ **True** □ **False**

3

False.
Monte carlo method involves generation of random numbers. Therefore, estimate of π after 10,000 iterations need not be closer to real value of π than 9000 iterations. It depends on the random numbers chosen during an execution.

Question 2 (5 marks) Explain how you will use computer simulations to determine the probability of rolling a 10 with two fair dice (preferably write Python code). □W □R

```
event = 0
iter = 10,000                    # no. of iterations can be changed.
for i in iter:
    a= randint (1,6)             # first roll
    b= randint (1,6)             # second roll
    if a+b==10 :
        event +=1

prob = event / iter              # computes probability
print (prob)
```

Question 3 (3 marks) Bootstrapping works well only when populations exhibit a normal distribution. □True □False

Question 4 (3 marks) Adding a +ve constant to every edge weight does not change the solution to the single-source shortest-paths problem □True □False

False

uestion 5 (3 marks) For representing very sparse graphs, adjacency matrix is not a very good data ructure

□ True □ False

True.
A sparse graph has ~~large~~ ~~more~~ very less no. of edges and hardly half of the matrix will be filled and the other half will be a waste of memory.

Question 6 (3 marks) In the Python package NetworkX, how will you find all nodes that connect to a given node "A"?

□ W □ R

G. get_nodes ("A")

Question 11 (4 marks) How many asterisks will be printed by the following code:

```
def fibo(n):
    print('*')

    if n == 1:
        return n
    elif n==2 or n==3:
        return n-1
    else:
        return fibo(n - 1) + fibo(n - 2)

def testFib(n):
    print ('fib of ', n, '=', fibo(n))

testFib(9)==testFib(10)
```

A 89   B 110   C 109   D 108   E 91

fib(9)

(8) + fib(7) —— * fib(6) + fib(5)

fib(7) + fib(6)

* fib(5) + fib(4)

fib(4) + fib(3) + fib(2)

fib(6) + fib(4) + fib(6)

fib(1) = *

fib(2) = *

fib(3) = *

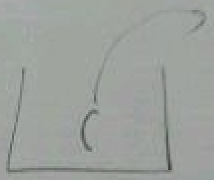
fib(4)

67
41
——
56

0 1 2 3 4

(3 + 5 2) 2 (4)

(



C

0 = 0

a b c d e f

f

d

c

a b

S

return n

return (fib 2, fib(3)) —— 1

~~[illegible]~~ list L

v: (

2

5 9 15 25 41 67 3

1 1 3 ~~[illegible]~~ ~~[illegible]~~ ~~[illegible]~~

2 3 4 5 6 7 8 9 10 4

= *

= *

) = 3

) = ~~[illegible]~~ 5

) = 10

) = 17

41
67
——
108

1 x 2

Question 3 (3 marks) The worst-case time complexity of randomised quicksort is $\Theta(n^2)$

□ False □ True

Question 4 (10 marks) What is the edit distance between the strings 'CUAUGA' and 'UAUGGA'? Show the table for computing the edit distance. Assume that the costs of a substitution, insertion and deletion are all unity.

t \ s	–	C	U	A	U	G	A
–	0	1	2	3	4	5	6
U	1	1	1	2	3	4	5
A	2	2	2	1	2	3	4
U	3	3	2	2	1	2	3
G	4	4	3	3	2	1	2
G	5	5	4	4	3	2	2
A	6	6	5	4	4	3	2

□ W □ PR □ MR □ FR

2

k=1

5191

35414
15434
13454

Question 1 (4 marks) What is a stable sorting algorithm? □W □R

A stable algorithm easily identifies a sorted information and does not monotonously perform unnecessary operations

0

Question 2 (3 marks) On already sorted data, selection sort will be faster than insertion sort.

□ True ☑ False

False. Insertion sort will be faster on sorted data than selection sort. Selection sort repeats through the ~~el~~ entire data comparing elements while insertion sort runs through the entire data once trying to fit numbers in their correct order and halts immediately

3

31445
13445

1
12 45
3145 4
3415 4
3412 4

Question 2 ♣ (2 marks) Which of the following are ways to initialise a list of empty lists in Python? The expected output is `[ [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ] ]`

A `x= [ ] * 10`

B `x= [None] * 10`

C `x= [ [ ] * 10]`

D `x = [ [ ] for i in range (10)]` ✓

E
```
x= [ ]
for i in range(10):
  x.append([])
```
✓

Question 3 ♣ (2 marks) The variable d has the value 95. Which of the following will return `True`?

A `d is 95.0`

B `d == 95.0` ✓

C `d is 95` ✓

D `d is not None` ✓

E `d == 95` ✓

Question 4 (1 mark) In the following code snippet, `data` and `rainfall` are:

```
def compute_average(data):
    """ Compute the average of a list of numbers
    """
    return sum(data)/len(data)

>>> print('Average rainfall: ',compute_average(rainfall))
```

A both function parameters

B both function arguments

C function parameter and function argument, respectively ✓

D function argument and function parameter, respectively

E None of the above

Question 5 ♣ (2 marks) Which of the following statements will print `'to-be-or-not-to-be'`:

A `print('to','be','or','not','to','be',sep='-')` ✓

B `print('to-'+'be-'+'or-'+'not-'+'to-'+'be')` ✓

C `print('-'.join(['to','be','or','not','to','be']))` ✓

D `print('to','be','or','not','to','be')`

E `print(['to','be','or','not','to','be'])`

Question 6 ♣ (2 marks) Which of the following are acceptable as dictionary keys in Python?

[✓ A] `'Hello world!'` [✓ B] `True` [✓ C] `3.0` [✓ D] `(1,2)` E `[1,2]`