IT5001 Midterm Exam Questions (AY2023/2024, SEM1)

Multiple Choice Questions (MCQs) 25 x 2 = 50 marks

You are given a complete Python program stored in a `.py` file. Determine the output (if any) of the program upon execution.

QN	Questions	Answer
1	`f = print` `print(type(f))`	a) `<class 'str'>` b) `<class 'NoneType'>` c) `<class 'function'>` d) `<class 'builtin_function_or_method'>` e) Error
2	`print( 1+-+1+-+1+-+1 )`	a) -2 b) 2 c) 1 d) -1 e) Error
3	`print( 0 * '1' )`	a) '0' b) '1' c) 0 d) '' e) Error
4	`print('1'*3+'2'*'4')`	a) '1112222' b) '1324' c) '38' d) '344' e) Error

5	`a, b, c, d = 'a','b','c','d'`
	`a, b, c, d = d, c, b, a`
	`print(a,b,c,d)`
		a) 'd c c d'
		b) 'd c b a'
		c) 'dccd'
		d) 'dcba'
		e) Error

6	`s = 'abc'`
	`s[1] = 'x'`
	`print(s)`
		a) 'abc'
		b) 'axc'
		c) 'axbc'
		d) 'abxc'

		e) Error
7	print('#')#')#')#')'[2])	a) ''' b) "'" d) ')' d) '#' e) Error
8	h = 2 h *= 2 h += 2 h >= 2 print(h)	a) True b) False c) 2 d) 6 e) Error

9	var = 1
	def p():
	    var = 2
	def q():
	    var = p()
	    return var
	q()
	print(var)
		a) 2
		b) 1
		c) 0
		d) None
		e) Error

10	d = (True, False, True, False) print(d[3:] == (d[3]))	a) True b) False c) None d) 3 e) Error
11	lst1 = [1,2] lst2 = [3,4] lst3 = [5,6] lst2.append(lst1) lst3.append(lst2) print(lst3)	a) [5, 6] b) [5, 6, [1, 2]] c) [5, 6, 3, 4, 1, 2] d) [5, 6, [1, 2, [3, 4]]] e) [5, 6, [3, 4, [1, 2]]]
12	lst1 = [i for i in range(1,4)] lst2 = [i for i in range(1,6,2)] print(len(set(lst1+lst2)))	a) 1 b) 2 c) 4 d) 6 e) Error
13	e = [1, 2, 3, 4, 5] print(e[:-2] + e[2:])	a) [1, 2, 3, 4, 5] b) [1, 2, 4, 5] c) [1, 2, 3, 3, 4, 5]

		d) [4, 6, 8] e) Error
14	f = {'IT', '500', 1} g = {'IT', '500', 3} print(f^g)	a) {1} b) {1, 3} c) {'IT', '500'} d) {'IT', '500', 1, 3} e) Error
15	i = True j = False k = {i: j} i = False print(k)	a) {True: False} b) {False: False} c) {True: False, False: False} d) Error e) None
16	print(tuple(list('set')))	a) (['s', 'e', 't'],) b) (['set'],) c) ('s', 'e', 't') d) ([{}],) e) ('set',)
17	b = [[2, 7], [0, 1]] c = b[:].copy() c[0][0] = 3 print(b)	a) [[2, 7], [0, 1]] b) Error c) [[2, 7], [3, 1]] d) [[3, 7], [0, 1]] e) [3, [0, 1]]
18	x = [1, 2] print(append(x, 3))	a) [1, 2, 3] b) [1, 2, [3]] c) [1, 2] d) None e) Error
19	x = ['i', 't', '5', '0', '0', '1'] y = [] for i in range(3): y.append(x.pop(i)) print(y)	a) ['i', 't', '5', '0'] b) ['i', 't', '5'] c) ['i', '5', '0'] d) ['i', 't'] e) []
20	print({1} + {1})	a) {{1}, {1}} b) {1, 1} c) {2} d) {1} e) Error
21	x = ((1, 2), (3, 4), (5, 6))	a) 1

	`print((lambda f: f(x))(len))`	b) 2 c) 3 d) 6 e) Error
22	`print((lambda x : lambda y: y ** x)(2)(3))`	a) 2 b) 3 c) 6 d) 9 e) Error
23	`x = [1] * 3` `print(1.0 in x)`	a) True b) False c) Maybe d) None e) Error
24	`dict1 = {(1,2):[3,4], (5,6):[7,8]}` `print(dict1[(5,6)][0])`	a) 3 b) 4 c) 7 d) 8 e) Error
25	`dict1 = {123:'apple', 345:'orange', 'apple':789}` `print(dict1['apple'])`	a) 123 b) 345 c) 789 d) 'apple' e) Error

Open-ended Questions (OEQs) 15+15+20 = 50 marks

Q26. Observe the following program fragment.

```
def foo(lst):
    if not lst:
        return lst
    lst2 = []
    for x in lst:
        if x != lst[0]:
            lst2.append(x)
    return lst[:1]+foo(lst2)
```

Assuming the argument to `foo` is a list, describe `foo`; or in other words, what does `foo` do?

Q27. You are given a complete Python program stored in a `.py` file. Determine the output (if any) of the program upon execution.

```
numbers = [1, 2]
animals = ["dog", "cat"]

output = []
for x in numbers:
    for y in animals:
        output.append((x,y))
print(output)
```

Q28.

Any integer larger than 1 can be "decomposed" into a product of prime numbers (i.e. its prime factors). For example, 21 = 3 x 7 and 12 = 2 x 2 x 3. (Note: 1 is not a prime number)

Assuming that `n` is an integer which is larger than 1, replace each blank with a valid Python expression/statement such that the function `prime_factorize(n)` prints out all the prime factors of `n` in nondecreasing order.

Only **correct** (for all blanks) and **working** (i.e. no syntax errors/program crashes) code will score you marks.

Template:

```
def prime_factorize(n):
    i = 2
    while _______:
        if ________:
            ______
            print(i)
        else:
            ______
```

Example run:

```
>>> prime_factorize(45)
3
3
5
```

IT5001 Midterm Exam Solutions (AY2023/2024, SEM1)

Multiple Choice Questions (MCQs) 25 x 2 = 50 marks

You are given a complete Python program stored in a `.py` file. Determine the output (if any) of the program upon execution.

QN	Questions	Answer
1	`f = print` `print(type(f))`	a) <class 'str'> b) <class 'NoneType'> c) <class 'function'> d) <class 'builtin_function_or_method'> e) Error
2	`print( 1+-+1+-+1+-+1 )`	a) -2 b) 2 c) 1 d) -1 e) Error
3	`print( 0 * '1' )`	a) '0' b) '1' c) 0 d) '' e) Error
4	`print('1'*3+'2'*'4')`	a) '1112222' b) '1324' c) '38' d) '344' e) Error

5	`a, b, c, d = 'a','b','c','d'`
	`a, b, c, d = d, c, b, a`
	`print(a,b,c,d)`
		a) 'd c c d'
		b) 'd c b a'
		c) 'dccd'
		d) 'dcba'
		e) Error

6	`s = 'abc'`
	`s[1] = 'x'`
	`print(s)`
		a) 'abc'
		b) 'axc'
		c) 'axbc'
		d) 'abxc'
		e) Error

7	`print('#')#')#')#')'[2])`	a) `'''` b) `"'"` d) `')'` d) `'#'` e) Error

8	`h = 2`
	`h *= 2`
	`h += 2`
	`h >= 2`
	`print(h)`
		a) True
		b) False
		c) 2
		d) 6
		e) Error

9	`var = 1`
	`def p():`
	`  var = 2`
	`def q():`
	`  var = p()`
	`  return var`
	`q()`
	`print(var)`
		a) 2
		b) 1
		c) 0
		d) None
		e) Error

10	`d = (True, False, True, False)` `print(d[3:] == (d[3]))`	a) True b) False c) None d) 3 e) Error

11	`lst1 = [1,2]`
	`lst2 = [3,4]`
	`lst3 = [5,6]`
	`lst2.append(lst1)`
	`lst3.append(lst2)`
	`print(lst3)`
		a) [5, 6]
		b) [5, 6, [1, 2]]
		c) [5, 6, 3, 4, 1, 2]
		d) [5, 6, [1, 2, [3, 4]]]
		e) [5, 6, [3, 4, [1, 2]]]

12	`lst1 = [i for i in range(1,4)]`
	`lst2 = [i for i in range(1,6,2)]`
	`print(len(set(lst1+lst2)))`
		a) 1
		b) 2
		c) 4
		d) 6
		e) Error

13	`e = [1, 2, 3, 4, 5]` `print(e[:-2] + e[2:])`	a) [1, 2, 3, 4, 5] b) [1, 2, 4, 5] c) [1, 2, 3, 3, 4, 5] d) [4, 6, 8]

		e) Error
14	f = {'IT', '500', 1} g = {'IT', '500', 3} print(f^g)	a) {1} b) {1, 3} c) {'IT', '500'} d) {'IT', '500', 1, 3} e) Error
15	i = True j = False k = {i: j} i = False print(k)	a) {True: False} b) {False: False} c) {True: False, False: False} d) Error e) None
16	print(tuple(list('set')))	a) (['s', 'e', 't'],) b) (['set'],) c) ('s', 'e', 't') d) ([{}],) e) ('set',)
17	b = [[2, 7], [0, 1]] c = b[:].copy() c[0][0] = 3 print(b)	a) [[2, 7], [0, 1]] b) Error c) [[2, 7], [3, 1]] d) [[3, 7], [0, 1]] e) [3, [0, 1]]
18	x = [1, 2] print(append(x, 3))	a) [1, 2, 3] b) [1, 2, [3]] c) [1, 2] d) None e) Error
19	x = ['i', 't', '5', '0', '0', '1'] y = [] for i in range(3): y.append(x.pop(i)) print(y)	a) ['i', 't', '5', '0'] b) ['i', 't', '5'] c) ['i', '5', '0'] d) ['i', 't'] e) []
20	print({1} + {1})	a) {{1}, {1}} b) {1, 1} c) {2} d) {1} e) Error
21	x = ((1, 2), (3, 4), (5, 6)) print((lambda f: f(x))(len))	a) 1 b) 2

		c) 3 d) 6 e) Error
22	`print((lambda x : lambda y: y ** x)(2)(3))`	a) 2 b) 3 c) 6 d) 9 e) Error
23	`x = [1] * 3` `print(1.0 in x)`	a) True b) False c) Maybe d) None e) Error
24	`dict1 = {(1,2):[3,4], (5,6):[7,8]}` `print(dict1[(5,6)][0])`	a) 3 b) 4 c) 7 d) 8 e) Error
25	`dict1 = {123:'apple', 345:'orange', 'apple':789}` `print(dict1['apple'])`	a) 123 b) 345 c) 789 d) 'apple' e) Error

Open-ended Questions (OEQs) 15+15+20 = 50 marks

Q26. Observe the following program fragment.

```
def foo(lst):
    if not lst:
        return lst
    lst2 = []
    for x in lst:
        if x != lst[0]:
            lst2.append(x)
    return lst[:1]+foo(lst2)
```

Assuming the argument to `foo` is a list, describe `foo`; or in other words, what does `foo` do?

`foo` returns a list of unique elements from `lst` without the duplicates .

Q27. You are given a complete Python program stored in a `.py` file. Determine the output (if any) of the program upon execution.

```
numbers = [1, 2]
animals = ["dog", "cat"]

output = []
for x in numbers:
    for y in animals:
        output.append((x,y))
print(output)
```

```
[(1, 'dog'), (1, 'cat'), (2, 'dog'), (2, 'cat')]
```

Q28.

Any integer larger than 1 can be “decomposed” into a product of prime numbers (i.e. its prime factors). For example, 21 = 3 x 7 and 12 = 2 x 2 x 3. (Note: 1 is not a prime number)

Assuming that `n` is an integer which is larger than 1, replace each blank with a valid Python expression/statement such that the function `prime_factorize(n)` prints out all the prime factors of `n` in nondecreasing order.

Only **correct** (for all blanks) and **working** (i.e. no syntax errors/program crashes) code will score you marks.

Template:

```
def prime_factorize(n):
    i = 2
    while _______:
        if ________:
            ______
            print(i)
        else:
```

Example run:

```
>>> prime_factorize(45)
3
3
5
```

```
def prime_factorize(n):
    i = 2
    while n != 1:
        if n%i==0:
            n//=i
            print(i)
        else:
            i+=1
```