

06 පහත දක්වා තිබෙන ක්‍රමලේඛ සම්බන්ධයෙන් ප්‍රකාශණ තුනක් ක්‍රමලේඛ කේතය සමඟ දක්වා ඇත.

A.	B.	C.
<pre>balance = 10 while True: if balance < 9: continue balance = balance - 9 print(balance)</pre>	<pre>balance = 10 while True: if balance < 9: break balance = balance - 9 print(balance)</pre>	<pre>balance = 10 while True: if balance < 9: pass balance = balance - 9 print(balance)</pre>

- A - 1 අගය ප්‍රතිදානය කර Loop අඛණ්ඩව ක්‍රියාත්මක කරයි.
- B - ප්‍රතිදානය ලෙස 1 මුද්‍රණය කර ක්‍රියාවලිය නතර කරයි.
- C - සෘණ අගයයන් පමණක් මුද්‍රණය කර Loop අඛණ්ඩව ක්‍රියාත්මක කරයි.

නිවැරදි ප්‍රකාශණ කුමක් වේ ද?

- (1) A පමණක් සත්‍ය වේ.
- (2) A සහ B සත්‍ය වේ.
- (3) B සහ C සත්‍ය වේ.
- (4) A සහ C සත්‍ය වේ.
- (5) A, B හා C සත්‍ය වේ.

07 පහත ක්‍රමලේඛය ක්‍රියාත්මක කරමින් 10 හා 20 ආදාන අගයන් ලෙස පිළිවලින් ලබාදුන් විට ලැබෙන ප්‍රතිදානය කුමක් වේ ද?

```
first = input("Enter first number :")
second = input("Enter second number :")
print(first * second)
```

- (1) 100
- (2) 200
- (3) 10
- (4) 20
- (5) TypeError දෝෂය ප්‍රතිදානය කරයි.

08 list1 = [1, 3] ක්‍රමලේඛය ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිදානය කුමක් වේ ද?

```
list1.append(5)
list2 = list1
list1.reverse()
print(list1)
print(list2)
```

- (1) [1, 3, 5]
- (2) [5, 3, 1]
- (3) [1, 3, 5]
- (4) [1, 3, 5]
- (5) [5, 3, 1]
- (6) [5, 3, 1]
- (7) ඉහත කිසිවක් සත්‍ය නොවේ.

09 පයිතන් ක්‍රමලේඛ භාෂාවේ සතු ලක්ෂණයක් නොවන්නේ කුමක් ද?

- (1) වස්තු නැඹුරු භාෂාවකි.
- (2) පරිවර්තක මෘදුකාංගය සම්පාදක (Compiler) වේ.
- (3) විවෘත මෘදුකාංගයකි.
- (4) සරල වාග් රීති සහිත භාෂාවකි.
- (5) ඉගැන්වීමට පහසු භාෂාවකි.

10 >>> round(2**6 + 3/2 + 12%5) ප්‍රකාශණයෙහි අගය වන්නේ

- (1) 15
- (2) 67.5
- (3) 67
- (4) 35
- (5) 68

11 >>> "Good " + 1 + 2 + 3 ප්‍රකාශණයෙහි පවතින දෝෂය ආකාරය වන්නේ කුමක් ද?

- (1) TypeError
- (2) NumberError
- (3) IndexError
- (4) KeyError
- (5) SyntaxError

- 12 `data = ['ab', 'cd']` නිවැරදි ප්‍රතිදානය කුමක් වේ ද?
`for d in data:`
`d.upper()`
`print(data)`
- (1) ['ab', 'cd'] (2) ['AB', 'CD']
(3) 'AB' 'CD' (4) AB CD
(5) ab cd
- 13 පයිතන් ක්‍රමලේඛ භාෂාවේ විචල්‍ය (Variable) භාවිතය අනුව නිවැරදි ප්‍රකාශය කුමක් ද?
A. විචල්‍ය භාවිතයට පෙර අගය සමඟ අර්ථ දැක්වීම කළයුතු වේ.
B. ක්‍රියාපටිපාටි තුළ භාවිතා වන විචල්‍යයන් එම සීමාව තුළ පමණක් ක්‍රියාත්මක වේ.
C. සෑම විචල්‍යයක පවතින අගය ක්‍රමලේඛය තුළ ඕනෑම පේළියක දී භාවිතා කළ හැකි වේ.
- (1) A පමණක් සත්‍ය වේ. (2) B,C පමණක් සත්‍ය වේ.
(3) B පමණක් සත්‍ය වේ. (4) A, B පමණක් සත්‍ය වේ.
(5) A,B,C සත්‍ය වේ.
- 14 සිසුන් ලබාගත් ලකුණු අනුව ඉසෙඩ් අගය ගණන කිරීමේ දී, ලකුණු තබා ගැනීම සඳහා භාවිතා කළ යුතු දත්ත ව්‍යුහය වන්නේ
(1) List (2) Tuple
(3) Dictionary (4) string
(5) Text file
- 15 `>>> x = ((1, 2), (3, 4, 5), (5, 6, 5, 9))`
`>>> len(x)` ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිදානය කුමක් වේ ද?
(1) 3 (2) 9
(3) 2 (4) 4
(5) 8
- 16 දෝෂ ආකාරයක් නොපවතින දත්ත ව්‍යුහය කුමක් වේ ද?
(1) numbers = [1:'One', 2:'Two', 3:'Three'] (2) numbers = (1:'One', 2:'Two', 3:'Three')
(3) numbers = {1:One, 2:Two, 3:Three} (4) numbers = {1:'One', 2:'Two', 3:'Three'}
(5) numbers = 1:'One', 2:'Two', 3:'Three'
- 17 `print("A", end = 'x')` ලැබෙන ප්‍රතිදානය කුමක් ද?
`print("B", sep = ', ', end = '')`
`print("C", sep = ', ', end = '')`
`print("D", sep = ', ', end = '.')`
- (1) AxB,C,D.
(2) AxB,C,D
(3) ABCD
(4) AxBCD.
(5) A,x,B,C,D.
- 18 පහත ක්‍රමලේඛය ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිදානය වන්නේ
`matrix = [[1, 2, 3, 4],`
`[5, 6, 7, 8],`
`[9, 10, 11, 12],`
`[13, 14, 15, 16]]`
`for i in range(0, 4):`
`print(matrix[i][1], end = " ")`
- (1) 2 6 10 14
(2) 1 2 3 4
(3) 4 8 12 16
(4) 1 5 9 13
(5) 13 12 15 16
- 19 `>>> st = 'KATARAGAMA'`
`>>> st.strip().split('A')` ලැබෙන ප්‍රතිදානය වන්නේ කුමක් ද?
(1) ['K', 'T', 'R', 'G', 'M', ''] (2) ['K', 'T', 'R', 'G', 'M']
(3) ['K', 'A', 'T', 'A', 'R', 'A', 'G', 'A', 'M', 'A'] (4) ['K', 'A', 'T', 'A', 'R', 'A', 'G', 'A', 'M', 'A']
(5) ඉහත කිසිවක් සත්‍ය නොවේ.

- 20 `def nPrint(message, n):`
`while n > 0:`
`print(message, end='')`
`n -= 1`
- පයිතන් ක්‍රමලේඛය අනුව,
`nPrint(n = 2, message = "python")`
- උපදේශය ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිදානය කුමක් ද?
- (1) python
 python
 (2) pythonpython
 (3) Python python
 (4) "python"
 (5) SyntxaError
- 21 ක්‍රියාපරිපාටිය තුළ ප්‍රත්‍යාගමන (return) අගයක් සඳහන් කර නොමැති විටක return වන්නේ කුමක් ද?
- (1) None (2) int
 (3) String (4) null
 (5) ඉහත කිසිවක් නොවේ.
- 22 `x = 1` ප්‍රතිදානය කුමක් ද?
`def f1():`
`x = 5`
`print(x)`
- `f1()`
`print(x)`
- (1) 5 (2) 5
 1 5
 (3) 1 (4) 1
 1 5
 (5) දෝෂ ආකාරයක් ප්‍රතිදානය කරයි.
- 23 `n = 0` ක්‍රමලේඛය ක්‍රියාත්මක වීම අනුව අගය ප්‍රතිදානය වාර ගණන වන්නේ
`m = n`
`while n < 10:`
`while m < 20:`
`m = m + 1`
`print(m,n)`
`n = n + 1`
- (1) 20 (2) 30
 (3) 200 (4) 10
 (5) ඉහත කිසිවක් නිවැරදි නොවේ.
- 24 Dictionary දත්ත ව්‍යුහයන් අනුව පහත පයිතන් ප්‍රකාශනය ක්‍රියාත්මක කළ විට ලැබිය හැකි ප්‍රතිදානය කුමක් ද?
- `a1 = {1:10,2:20}`
`a2 = {3:30}`
`print(a2 + a1)`
- (1) {1:10,2:20,3:30}
 (2) {3:30,1:10,2:20}
 (3) {4:40,5:50}
 (4) TypeError ප්‍රතිදානය කරයි.
 (5) ඉහත කිසිවක් සත්‍ය නොවේ.
- 25 `def f1(x = 1, y = 2):` ක්‍රමලේඛය ක්‍රියාත්මක කළ විට ප්‍රතිදානය කුමක් වේ ද?
`x = x + y`
`y += 1`
`print(x, y)`
- (1) 3 3 (2) 1 2
 (3) 3 4 (4) 3 2
 (5) ඉහත කිසිවක් නොවේ.
- 26 හාග සංඛ්‍යාවක් නිරූපණය වන්නේ කුමන ප්‍රකාශය මගින් ද?
- (1) 2E2 (2) 0b101
 (3) 0xCD (4) 0o34
 (5) ඉහත කිසිවක් නොවේ.
- 27 පයිතන් ක්‍රමලේඛ තුළ ක්‍රියාපරිපාටි (functions) භාවිතයේ වාසියක් වන්නේ කුමක් ද?
- (1) උපදෙස් කාණ්ඩ නැවත නැවත අඩංගු කිරීම වැළැක්වීම
 (2) සංකීර්ණ ක්‍රියාවලියක් සරල කොටස් බවට පත්කිරීමේ හැකියාව
 (3) කේත කාණ්ඩ නැවත භාවිතය (Reuse of code)
 (4) ක්‍රියාවලියක් සිදුවන උපදෙස් පරිශීලකයාගෙන් සැඟවීම (Information Hiding)
 (5) ඉහත සියල්ලම නිවැරදි වේ.

- 28 ගොණු ක්‍රියාත්මක පිහිටුම ලබාගැනීම සඳහා භාවිතා වන ක්‍රියාපටිපාටියක් විය හැක්කේ කුමක් ද?
- (1) seek() (2) tell()
 (3) index() (4) read()
 (5) readlines()
- 29 `>>> 2 * (2 ** 3) % 3 + 12` උපදේශය ක්‍රියාත්මක කළ විට ලැබුණ ප්‍රතිදානය ම ලැබෙන පයිතන් ප්‍රකාශය කුමක් වේ ද?
- (1) `>>> 2 * (2 ** 3) / 3 + 12` (2) `>>> 2**4 % 3 + 12`
 (3) `>>> 4 ** 3 / 15` (4) `>>> 4 ** 3 / 3 + 12`
 (5) `>>> 2 * 2 ** 3 % 2 + 12`
- 30 `>>> data = ['A','B','C']`
`>>> data.extend([10,20])` ක්‍රියාත්මක කළ විට තිරය මත ලැබෙන ප්‍රතිදානය කුමක් වේ ද?
- (1) ['A', 'B', 'C', 10, 20] (2) ['A', 'B', 'C', [10, 20]]
 (3) [10, 20, 'A', 'B', 'C'] (4) දෝෂයක් ප්‍රතිදානය කරයි.
 (5) කිසිවක් ප්‍රතිදානය නොකරයි.
- 31 ක්‍රියාපටිපාටිය අර්ථ දැක්වීම නිවැරදිව දක්වන ප්‍රකාශය කුමක් ද?
- (1) `def f(a = 1, b):`
 (2) `def(a = 1, b, c = 2):`
 (3) `def f(a = 1, b = 1, c = 2):`
 (4) `def f(a = 1, b = 1, c = 2, d):`
 (5) `def f(a = 1, b = 1, c = 2, d):`
- 32 `temperature = 50` පයිතන් තුළ භාවිතා වන loop සම්බන්ධයෙන් නිවැරදි නොවන්නේ කුමක් ද?
- `if temperature >= 100:` (1) `too hot`
`print("too hot")` (2) `too cold`
`elif temperature <= 40:` (3) `just right`
`print("too cold")` (4) කිසිවක් මුද්‍රණය නොවේ.
`else:` (5) දෝෂ ආකාරයක් දිස් වේ.
`print("just right")`
- 33 පහත පයිතන් (python) ක්‍රමලේඛ අනුව වඩාත් ම නිවැරදි ප්‍රකාශය කුමක් ද?
- `even = True` (1) `It is even!`
`if even:` (2) `It is not even!`
`print("It is even!")` (3) කිසිවක් දිස් නොවේ.
`else` (4) `if` තර්කනය වැරදි බැවින් `Error` දිස් වේ.
`print("It is not even!")` (5) `SyntaxError`
- 34 `>>> d1 = ['4', '6']` ප්‍රතිදානය වන්නේ කුමක් ද?
`>>> d2 = [5,10]`
`>>> d1 + d2` (1) [4, 6, 5, 10] (2) [8, 12]
 (3) ['4', '6', 5, 10] (4) ['4', '6', '4', '6']
 (5) [9, 16]
- 35 `num = 10/3` නම් දෝෂ ආකාරයක් ප්‍රතිදානය වන ප්‍රකාශය කුමක් ද?
- (1) `eval(num)` (2) `int(num)`
 (3) `round(num)` (4) `round(num,2)`
 (5) `str(num)`
- 36 දෛනිකව ලබාගන්නා වර්ෂාපතන අගයයන් ගොණුව (`rain.txt`) තුළ සටහන් කිරීමේ දී, එම ගොණුව විවෘත කළ යුතු උපදේශය නිවැරදිව දැක්වෙන්නේ
- (1) `rain = open('rain.txt', 'a')` (2) `rain = open('rain.txt', 'r')`
 (3) `rain = open('rain.txt', 'w')` (4) `rain = open('rain.txt', 'w+')`
 (5) `rain = open('rain.txt', 'a+')`

37 `count = 0` පයිතන් ක්‍රමලේඛය අනුව නිවැරදි ප්‍රකාශය තෝරන්න.
`# Point A`
`for count in range(0,10):`
 `print(count)`
 `count += 1`
`# Point B`
`# Point C`

(1) Point A තුළ දී සෑම විටම `count < 0` වේ.
(2) Point B තුළ දී සෑම විටම `count > 0` වේ.
(3) Point B තුළ දී සෑම විටම `count < 0` වේ.
(4) Point C තුළ දී සෑම විටම `count < 0` වේ.
(5) Point C තුළ දී සෑම විටම `count = 10` වේ.

38 List දත්ත ව්‍යුහය අනුබද්ධයෙන් නිවැරදි ප්‍රකාශය කුමක් ද?

- (1) දත්ත ඉවත් කිරීම සඳහා `remove()` පමණක් භාවිතා කරයි.
- (2) දත්ත ඇතුළු කිරීම සඳහා `insert()` පමණක් භාවිතා කරයි.
- (3) දත්ත වෙනස් කිරීම සඳහා `pop()` භාවිතා කළ හැක.
- (4) අගයයක් ඉවත් කිරීම සඳහා `del` භාවිතා කළ හැකිය.
- (5) ඉහත කිසිවක් නිවැරදි නොවේ.

39 `sum = 0` ක්‍රමලේඛය ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිදානය කුමක් වේ ද?
`item = 0`
`while item < 5:`
 `item += 1`
 `sum += item`
 `if sum > 4:`
 `break`
 `print(sum)`

(1) 5 (2) 6
(3) 8 (4) 7
(5) කිසිවක් ප්‍රතිදානය නොකරයි.

40 පයිතන් ක්‍රමලේඛ භාෂාවේ නිර්මාතෘවරයා වන්නේ කවුරුන් ද?

- (1) James Gosling (2) Bjarne Stroustrup
- (3) John McCarthy (4) Guido van Rossum
- (5) Ada Augusta Lovelace

41 `>>> print('num1 {0}, num2 {2}, num3 {2}'.format(10, 15, 20))`

ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිදානය කුමක් ද?

- (1) num1 10, num2 15, num3 20 (2) num1 10, num2 15, num3 15
- (3) num1 10, num2 20, num3 20 (4) num1 20, num2 15, num3 10
- (5) num1 15, num2 10, num3 20

42 ක්‍රමලේඛ හෝ ක්‍රියා පටිපාටි තුළ දර්ථ දක්වන විචල්‍යය ආකාරයක් නොවන්නේ කුමක් ද?

- (1) ආවේණික පරාමිත (2) අනිවාර්ය පරාමිති
- (3) ස්ථානීය විචල්‍යය (4) විශ්ව පරාමිති
- (5) මූල පද තර්ක

43 අරය, `radius` සඳහා ආදානය කළ විට වෘත්තයක වර්ගඵලය ගණනය කිරීම නිවැරදිව දැක්වෙන ප්‍රකාශය කුමක් ද?

- (1) `if radius != 0: Print(radius * radius * 3.14159)`
- (2) `if radius >= 0: Print(radius * radius * 3.14159)`
- (3) `if radius > 0: Print(radius * radius * 3.14159)`
- (4) `if radius <= 0: Print(radius * radius * 3.14159)`
- (5) ඉහත කිසිවක් නොවේ.

44 `n = 0` පයිතන් loop ක්‍රියාත්මක වීම අනුව n සඳහා ලැබෙන අගය කුමක් ද?
`for i in range(5,10):`
 `for j in range(0,i):`
 `n = n+1`
`print(n)`

(1) 35 (2) 5
(3) 11 (4) 18
(5) 26

45 පයිතන් උපදෙස් කාණ්ඩය ක්‍රියාත්මක කිරීමේ දී ප්‍රතිදානය වන දෝෂ ආකාරය සම්බන්ධයෙන් නිවැරදි ප්‍රකාශය කුමක් ද?

```
>>> a = 0
>>> c = 10
>>> d = a / (c - 10)
```

Traceback (most recent call last):
File "<pyshell#11>", line 1, in <module>
d = a / (c - 10)
ZeroDivisionError: integer division or modulo by zero

- (1) උපදෙස් ක්‍රියාත්මක කිරීමේ දී තාර්කික දෝෂ ආකාරයක් අන්තර්ගත වේ.
- (2) C සඳහා අගය ආදේශ කිරීම අනුව දෝෂ ආකාරය නිර්මාණය වී ඇත.
- (3) හරය ශුන්‍ය බැවින් වාග්මිති දෝෂ (SyntaxError) ආකාරය නිර්මාණය වී ඇත.
- (4) (c - 10) උපදෙස් ක්‍රියාත්මක කිරීමේ දී දෝෂය නිර්මාණය වී ඇත.
- (5) ඉහත කිසිවක් සත්‍ය නොවේ.

46 'marks.txt' ගොනුවක් කියවීම සඳහා භාවිතා වන පහත දැක්වෙන කුමන පයිතන් (Python) ක්‍රමලේඛ කණ්ඩය කාරක නීති අනුව නිවැරදි (syntactically correct) වේද?

- (1)

```
file = open('marks.txt', 'a')
data = file.readlines()
for line in data:
    print(line)
```
- (2)

```
file = open('marks.py', 'r')
data = file.readlines()
for line in data:
    print(line)
```
- (3)

```
file = open('marks.txt', 'w')
data = file.readlines()
for line in data: print(line)
```
- (4)

```
file = open('marks.txt', 'r')
data = file.readlines()
for line in data:
    print(line)
```
- (5)

```
file = open('marks.txt', 'r')
data = file.readlines():
for line in data:
    print(line);
```

47 0b101 + 0x11 + 11 හි අගය නිවැරදි ව නිරූපණය වන්නේ

- (1) 27
- (2) 33
- (3) 123
- (4) 129
- (5) ඉහත කිසිවක් නිවැරදි නොවේ.

48 පයිතන් while loop භාවිතය සම්බන්ධව අසත්‍ය ප්‍රකාශය කුමක් ද?

- (1) දත්ත වාර ගණනක් උපදෙස් ක්‍රියාත්මක කිරීම සඳහා භාවිතා කළ හැකිය.
- (2) තර්කයක් සත්‍ය වන තෙක් උපදෙස් නැවත නැවත ක්‍රියාත්මක කිරීම භාවිතා කළ හැකිය.
- (3) ඇතම් while loop ප්‍රකාශන, for loop භාවිතයෙන් එම ප්‍රතිදානය ම ලැබෙන පරිදි සකස් කළ හැකිය.
- (4) ක්‍රියාත්මක වන while loop සමඟ ක්‍රියාවලිය අවශ්‍ය විටක නතර කළ හැකිය.
- (5) අපිරිමිත පුනරාවර්තන (infinite Loop) සඳහා උපදෙස් පෙළ ගැස්විය හැක.

49 ආදානය කළ හැඳුනුම්පත් අංක (nid) නිවැරදි දැයි පරීක්ෂාව සඳහා භාවිතා කළ නොහැකි උපදෙස් කාණ්ඩය කුමක් ද?

- (1) n = len(nid)
- (2) if not(n == 10):
- (3) nid = input('NID:')
- (4) elif nid[-1] != 'V':
- (5) ඉහත කිසිවක් නිවැරදි නොවේ.

50 පහත ක්‍රමලේඛය ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිදානය කුමක් ද?

```
n = 6
for x in range(1,n):
    for y in range(x,n):
        print(y,sep='', end='')
    print('')
```

(1) 12345
2345
345
45
5

(4) 11111
2222
333
44
5

(2) 11111
2222
333
44
5

(5) 65432
6543
654
65
6

(3) ඉහත කිසිවක් සත්‍ය නොවේ.

