# 파이썬프로그래밍 시험 문제

## # 난이도 : 하

### 1) 다음을 실행했을 때 출력되는 값은?

|  |
| --- |
| 1e+02 - 3 + 2 \* 3 \*\* 2 |

1) 8.0 2) 115.0 3) 133.0 4) 1e+02 + 33 5) 1e+02 + 15

### **2)** x+y 의 값을 고르시오.(단, x > 0 , y > 0)

|  |
| --- |
| **a** = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] **print**(a[x:y])  결과: [6, 7, 8] |

1) 11 2) 12 3) 13 4) 14 5) 15

### **3)** ? 에 들어가기에 알맞은 것은?

|  |
| --- |
| for i in range(10):   if ? :   print(i)   결과 :  0  2  4  6  8 |

1) i == 0 2) i % 2 == 0 3) True 4) i%2 5) i % 2 = 0

### 

### 

### **4)** b[5] = ?

|  |
| --- |
| **b** = [ i\*2 **for** **i** **in** range(10)] |

1) 0 2) 1 3) 4 4) 7 5) 10

### 5) count에 최종적으로 들어있는 값은?

|  |
| --- |
| **data** = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] count = 0 for d **in** **data**:  **if** d%10 == 0:  count += 1 |

1) 4 2) 3 3) 2 4) 1 5) 0

### 6) 다음 코드가 무한루프를 돌지 않기 위한 ? 에 적절한 값은?

|  |
| --- |
| **while** ? :  print(1) |

1) 1 2) 'a' 3) True 4) 10 5) 0

### 7) len(data) = ?

|  |
| --- |
| data = [i\*10 **for** **i** **in** range(10)]  **len**(data) |

1) 10 2) 300 3) 1000 4) 3000 5) 5000

### 8) ? 에 들어가기에 알맞은 것을 쓰시오. 답 : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



|  |
| --- |
| **a** = [1, 2, 3, 4, 5] b = ['a', 'b', 'c', 'd', 'e']  **for** i, j **in** ? (**a**, b):  print(i, j)  결과 : 1, **a** 2, b 3, c 4, d 5, e |

### 9) ? 에 들어가기에 알맞은 것을 쓰시오. 답 : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| hero = ['아이언맨', '토르', '스파이더맨', '헐크'] hero. ? # 코드가 돌아갈 수 있도록 정확하게 모두 작성 요망 print(hero)  결과 :  ['아이언맨', '토르', '스파이더맨', '헐크', ‘그루트’, ‘블랙팬서’] |

### 10) ? 에 들어가기에 알맞은 단어를 쓰시오. 답 : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| 클래스의 재사용을 위해 부모 클래스로부터  **?** 하여 자식 클래스를 정의한다. |

## # 난이도 : 중

### 1) 다음을 실행했을 때 출력되는 값은?

|  |
| --- |
| count = 0  for i in range(10):  for j in range(5):  if i != j:  count += 1  else:  count = 0  print(count) |

1) 0

**2) 25**

3) 30

4) 45

5) 50

### 2~4) 다음은 문자열을 입력받아, 문자열이 모두 숫자로만 이루어져있는지 확인하는 코드입니다.

|  |
| --- |
| def check\_digit(key):  check = 0  if key in [str(i) for i in range(10)]:  *## check point (1) ##*  check = 1  return check  def is\_only\_digit(string):  for char in string:  if check\_digit(char):  continue  else:  return False  return True  is\_only\_digit('123') # - 결과 (1)  is\_only\_digit('ABC') # - 결과 (2)  is\_only\_digit('1A0') # - 결과 (3) |

### 2) 결과 (1) - 결과 (2) - 결과 (3)으로 알맞은 것은?

1. True - True - True 2) True - False - True 3) True - True - False

4) False - True - True **5) True - False - False**

### 3) 코드에서 주석처리된 check point (1)는 총 몇 회 수행되었을까?

1. 3회 **2) 4회** 3) 5회 4) 6회 5) 7회

### 4) 마지막에 check 변수에 들어가 있는 값으로 알맞은 것은?

1. **0** 2) 1 3) 2 4) 5 5) 10

### 5) 아래 코드를 수행한 결과로 알맞은 것은? (check에 들어가 있는 값)

|  |
| --- |
| company = ['Apple', 'Google', 'Adobe']  data = [comp.lower() for comp in company]  for d in data:  check = 0  if 'A' in d:  check += 1  print(check) |

1. **0**  2) 1 3) 2 4) 3 5) 16

### 6) 아래 코드에서 결과 (1)~(4) 로 알맞은 것은?

|  |
| --- |
| class BaseClassifier():  def \_\_init\_\_(self, max\_iter=1000, shuffle=True):  self.max\_iter = max\_iter  self.shuffle = shuffle   def fit(self):  print('Train')    def score(self):  print('Evaluate')   class Perceptron(BaseClassifier):  def \_\_init\_\_(self, alpha=0.001, max\_iter=10000, shuffle=False):  super().\_\_init\_\_(max\_iter=max\_iter)  self.alpha = alpha  self.shuffle = shuffle   *### 객체 생성 #####* model = Perceptron(alpha=0.1)   model.alpha *# 결과 (1)* model.shuffle *# 결과 (2)* model.fit() *# 결과 (3)* model.score() *# 결과 (4)* |

1. 결과 (1) : 0.001 / 결과 (2) : False / 결과 (3) : Train / 결과 (4) : Evaluate
2. 결과 (1) : 0.001 / 결과 (2) : True / 결과 (3) : Train / 결과 (4) : Evaluate
3. **결과 (1) : 0.1 / 결과 (2) : False / 결과 (3) : Train / 결과 (4) : Evaluate**
4. 결과 (1) : 0.1 / 결과 (2) : True / 결과 (3) : AttributeError / 결과 (4) : AttributeError
5. 결과 (1) : 0.1 / 결과 (2) : False / 결과 (3) : AttributeError / 결과 (4) : AttributeError

**7) ? 에 들어가기에 알맞은 것을 쓰시오. 답 : \_if i%10 = 0\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| for i in range(1, 101):   **? :**  print(i)  else:  print(i, end=' ')  결과 :  1 2 3 4 5 6 7 8 9 10  11 12 13 14 15 16 17 18 19 20  21 22 23 24 25 26 27 28 29 30  31 32 33 34 35 36 37 38 39 40  41 42 43 44 45 46 47 48 49 50  51 52 53 54 55 56 57 58 59 60  61 62 63 64 65 66 67 68 69 70  71 72 73 74 75 76 77 78 79 80  81 82 83 84 85 86 87 88 89 90  91 92 93 94 95 96 97 98 99 100 |

**8) ? 에 들어가기에 알맞은 것을 쓰시오. 답 : \_’ ‘\*(4-i) + ‘\* ‘\*i\*(i+1)\_\_\_\_**

|  |
| --- |
| for i in range(5):  print( **?** )  결과 :    \*   \* \*   \* \* \*   \* \* \* \*   \* \* \* \* \* |

### 9**)** 무한루프를 이용해서, 1~10까지 중 5의 배수만 제외하고 합을 구하는 코드를 완성해보자. 답 : (A)\_가\_\_(B)\_다\_\_(C)\_나\_\_(D)\_라\_\_

|  |
| --- |
| sum, num = 0, 0  while 1:  ------------------   (A) ------------------   (B) ------------------   (C) ------------------   (D) ------------------  sum |

|  |  |  |  |
| --- | --- | --- | --- |
| 가 | 나 | 다 | 라 |
| if num == 10:  break | if num%5 == 0:  continue | num += 1 | sum += num |

### 10) 다음 코드의 출력 결과를 쓰시오. 답 : \_16 ( = 8 + 7 + 1 )\_\_\_

|  |
| --- |
| data1 = [1, 2, 3, 4, 5] data2 = [4, 5, 6, 7, 8]  data3 = data1 + data2  data4 = set(data3)  data3 = list(data4)  data5 = [data3.pop(1)]  len(data3) + len(data4)+ len(data5) |

## 

## # 난이도 : 상

### 1) 아래는 문자열을 입력받아, 문자열이 모두 숫자로만 이루어져있는지 확인하는 코드입니다. 코드에서 주석처리된 check point (1)~(5)가 각각 수행된 횟수를 모두 더하면? 답\_11\_\_

|  |
| --- |
| def check\_digit(key):  check = 0  if key in [str(i) for i in range(10)]:  *## check point (1) ##*  check = 1   return check  def is\_only\_digit(string):  for char in string:  *## check point (2) ##*  if check\_digit(char):  *## check point (3) ##*  continue  else:  *## check point (4) ##*  return False  *## check point (5) ##*  return True  is\_only\_digit('010-0000-0000') |