

1) 파이썬 개발 도구(IDE)에 해당하는 것을 모두 고르시오.

- A. IDLE
- B. Colab
- C. WordPad
- D. PyCharm
- E. VSCode

2) `na = "3", nb = "5"`를 입력하면 출력 결과는?

```
na = input("숫자 하나를 입력하세요: ")
nb = input("또 다른 숫자를 입력하세요: ")
print(na + nb)
```

3) 다음의 출력 결과는?

```
a = 10
b = 5
```

```
if a > 5 and b < 2 or b == 5:
    print("조건 만족")
else:
    print("조건 불만족")
```

- A. 조건 만족
- B. 조건 불만족
- C. 에러 발생
- D. 아무 것도 출력되지 않음

4) 다음 파이썬 코드의 실행 결과로 옳은 것을 고르시오.

```
data = {
    "info": ("Python", [1, 2, 3])
}
```

```
data["info"][1].append(4)
data["info"][1][0] = 99
print(data["info"])
```

- A. ('Python', [99, 2, 3, 4])
- B. ('Python', [1, 2, 3, 4])
- C. ('Python', [1, 2, 3])
- D. 에러 발생 (튜플은 값을 변경할 수 없음)

5) 다음 중 **for**문과 **while**문, 그리고 **range()** 함수에 대한 설명으로 올바른 것을 모두 고른 것은?

- A. **for** 문은 리스트뿐만 아니라 문자열과 튜플 등 순서가 있는 자료형과 함께 사용할 수 있다.
- B. **range(3, 10, 2)**는 3부터 10까지 2씩 증가하는 숫자 3, 5, 7, 9를 생성한다.
- C. **while** 문은 조건이 **True**인 동안 반복되며, 반복 횟수가 정해진 경우에 적합하다.
- D. **break**문은 반복문을 즉시 종료하며, **continue**문은 다음 반복으로 넘어간다.

6) 다음 코드를 실행했을 때 출력값을 적으시오.

```
def add_to_list(value, target=[]):
```

```
    target.append(value)
```

```
    return target
```

```
print(add_to_list(1))
```

```
print(add_to_list(2))
```

```
print(add_to_list(3, []))
```

```
print(add_to_list(4))
```

7) 다음 코드의 출력 결과값으로 알맞은 것을 고르시오

```
a = 1
```

```
def change():
```

```
    print("1:", a)
```

```
    a = a + 1
```

```
    print("2:", a)
```

```
change()
```

A.

1: 1

2: 2

B.

1: 1

2: 1

C.

UnboundLocalError가 발생한다.

D.

NameError가 발생한다.

8) 선택정렬 함수의 빈칸을 적으시오.

def selection_sort(data):

for i in range(_____):

min_idx = i

for j in range(_____, len(data)):

if data[j] < data[min_idx]:

min_idx = j

data[i], data[min_idx] = _____

9) 다음의 출력 결과 값은?

```
global_count = 10

class Account:

    count = 5

    def __init__(self, balance):

        global global_count

        global_count += 1

        self.balance = balance + global_count + Account.count

        Account.count += 2

    def deposit(self, amount):

        self.balance += amount + Account.count

        Account.count += self.balance

        return self.balance

d

def double(x):

    return x * 2

a1 = Account(100)

a2 = Account(200)

x = a1.deposit(double(Account.count))

y = a2.deposit(x)
```

```
print("Global count:", double(1))

print("Account.count:", Account.count)

print("a1.balance:", a1.balance)

print("a2.balance:", a2.balance)

print("x:", x)

print("y:", y)
```

10) Car Class를 생성 , Car 를 부모 클래스로 하는 Bicycle 생성 , info 함수는 오버라이딩 하여

다음 코드에서 아래와 같이 출력되도록 하시오.

```
bicycle = Bicycle(2021, 2, 100, "시마노")
```

```
bicycle.info()
```

출력:

```
year : 2021
```

```
wheel : 2
```

```
price : 100
```

```
drivetrain : 시마노
```

11) tkinter에서 Tk() 클래스의 역할은?

- A. 버튼 위젯을 생성하는 클래스
- B. 최상위 윈도우 창(루트 창)을 생성하는 클래스
- C. 사용자 입력을 감지하는 이벤트 클래스
- D. 텍스트 출력을 위한 라벨 위젯 클래스

12) 다음 중 파이썬의 파일 처리 개념에 대한 설명으로 옳지 않은 것을 모두 고르시오.

- A. `open()` 함수는 파일이 존재하지 않으면 항상 에러를 발생시킨다.
- B. 'a' 모드로 파일을 열면 기존 내용을 삭제하고 새 내용을 기록한다.
- C. 'w' 모드로 파일을 열면 파일이 없을 경우 새로 생성된다.
- D. `with open(...) as f:` 문을 사용하면, 명시적으로 `f.close()`를 호출하지 않아도 된다.
- E. 텍스트 파일을 읽을 때 인코딩 방식을 지정하지 않으면 플랫폼 기본 인코딩 방식이 사용된다.
- F. `f.readlines()`는 파일의 모든 줄을 한 줄씩 읽어 리스트로 반환한다.

13) 다음 중 파이썬의 예외 처리, 문자열 처리, 람다 함수, `map` 함수에 대한 설명 중 옳지 않은 것을 모두 고르시오.

- A. `try...except` 구문에서 `except` 절은 여러 예외를 한 번에 처리할 수 있으며, 이때 튜플로 묶어 명시할 수 있다.
- B. `strip()` 메서드는 문자열의 앞뒤뿐 아니라 문자열 내부의 모든 공백을 제거한다.
- C. `lambda x: x + 1`는 `def f(x): return x + 1` 과 기능적으로 동일하다.
- D. `map()` 함수는 리스트뿐 아니라 문자열과 튜플에도 적용할 수 있다.
- E. `map()` 함수의 결과는 리스트가 아니라 이터레이터(iterator)로 반환된다.
- F. `try` 안에서 예외가 발생하지 않으면, `except` 블록은 실행되지 않는다.

14) 다음의 출력 결과 값은?

```
import re
```

```
text = "ID:1234, Name:홍길동, Email:hong@example.com"
```

```
pattern = r"ID:(\d+),\s+Name:([가-힣]+),\s+Email:([\w.-]+@[ \w.-]+\.\w+)"
```

```
match = re.search(pattern, text)
```

```
if match:
```

```
    print(match.group(1))
    print(match.group(2))
    print(match.group(3))
```

15) 다음 중 제너레이터(generator)에 대한 설명으로 옳지 않은 것은?

- A. `yield` 문이 있는 함수는 호출 시 제너레이터 객체를 반환한다.
- B. `next()` 함수를 호출할 때마다 함수는 중단된 위치에서 다시 실행된다.
- C. `return` 문은 제너레이터에서 값을 하나씩 반환하는 역할을 한다.
- D. `for` 문으로 제너레이터를 순회하면 내부적으로 `next()`가 자동 호출된다.
- E. 제너레이터는 메모리 효율적인 반복 처리를 위해 사용된다.

16) 문제: 다음 코드 실행 결과를 고르시오.

```
stack = []
stack.append(10)
stack.append(20)
stack.pop()
stack.append(30)
```

```
print(stack)
```

- A. [10, 20, 30]
- B. [10, 30]
- C. [30, 10]
- D. [20, 30]
- E. [10]

17) DFS(깊이 우선 탐색)의 특징으로 옳지 않은 것은?

- A. 스택 자료구조를 이용하여 구현할 수 있다.
- B. 한 노드를 방문하면 인접한 노드를 모두 방문하고 다음 단계로 넘어간다.
- C. 모든 노드를 방문하고자 할 때 주로 사용된다.
- D. 백트래킹(되돌아가기)을 활용하는 방식이다.
- E. 구현 시 재귀함수를 사용할 수도 있다.

18) 다음 코드를 실행하면 몇 각형이 그려지는가?

```
import turtle
t = turtle.Turtle()

for _ in range(6):
    t.forward(100)
    t.right(360/6)
```

- A. 정사각형
- B. 오각형
- C. 육각형

- D. 원
- E. 삼각형

19) 다음 NumPy 코드 실행 결과로 옳은 것은?

```
import numpy as np

a = np.array([1, 2, 3])
b = np.array([[1], [2], [3]])

result = a + b
print(result)
```

- A. 브로드캐스팅 오류 발생
- B. [2, 4, 6]
- C. [[2, 3, 4], [3, 4, 5], [4, 5, 6]]
- D. [[2], [4], [6]]
- E. [[1, 2, 3], [1, 2, 3], [1, 2, 3]]

20) 다음과 같은 선형 회귀 분석 결과가 주어졌을 때, 예측된 mpg 값이 가장 큰 데이터는 어느 것인가?

회귀식: $\text{mpg} = -0.07 \times \text{hp} + 30.1$

아래는 각 샘플의 마력(hp) 값이다.

샘플	hp
A	100
B	200
C	150

- A. A
- B. B
- C. C

D. A와 C

E. 모두 같다