

1장

1. 다음 중 **단순 자료구조**인 것을 고르시오.

색인파일, 정수, 리스트, 큐, 실수, 문자열, 트리, 그래프, 순차파일

2. 다음을 **선형 자료구조**와 **비선형 자료구조**로 구분하시오

리스트, 트리, 큐, 스택, 그래프

3. ()안에 들어갈 용어는?

(**순차 파일**)은 파일 내용을 논리적인 처리 순서에 따라 연속해서 저장하는 것이다. 즉, 입력되는 순서대로 저장한다고 보면 된다. 구조가 간단하기에 저장되는 공간 효율이 높지만, 다른 내용을 추가하거나 삭제할 경우에는 파일 내용을 재구성해야하므로 상당히 시간이 오래걸린다.

4. 어떤 용어에 대한 정의다. 빈칸에 들어갈 용어를 각각 채우시오.

(**자료구조**)는 컴퓨터 분야에서 효율적으로 접근하고 수정할 수 있도록 자료를 구성, 관리, 저장하는 것이다.

(**알고리즘**)은 컴퓨터나 수학 등 관련 분야에서 어떤 문제를 해결하기 위해 정해진 일련의 단계적인 절차나 방법을 공식화한 형태로 표현한 것이다.

5. 1~4의 설명에 해당하는 알고리즘 표현법을 고르시오.

보기 : 순서도로 표현, 일반 언어로 표현, 프로그램 언어로 표현, 의사코드 표현

①프로그램이나 알고리즘을 표현할 때, 프로그래밍 언어보다는 인간의 언어에 더 가까운 형태로 표현한 것. 프로그램 코드와 일반 언어의 중간형태 → **의사코드**

②여러 종류의 상자와 상자들이어주는 화살표를 통해 명령순서를 표현 → **순서도**

③일반사람이 이해하기 쉽지만, 최종적으로 코드로변경시 한계있음 → **일반언어**

④실제로 사용하는 프로그래밍언어의 코드로 바로 작성하는 것 → **프로그래밍언어**

3장

1. () 안에 들어갈 용어는?

(**선형 리스트**)는 데이터를 일정한 순서로 나열한 자료구조로, 입력 순서대로 저장하는 데이터에 적당하다.

2. 선형 리스트에 데이터를 삽입하는 과정 중 **거리가 먼 것은?**(정답:①)

①맨 앞에 공간을 하나 추가해야한다.

②마지막 위치에 바로 앞 위치의 데이터를 이동시킨다.

③삽입할 위치까지 ②를 반복한다.

④삽입할 위치에 데이터를 삽입한다.

3. 다음은 선형 리스트에 데이터를 삭제하는 과정이다. 순서대로 나열하시오.

(정답 : ④ → ② → ③ → ①)

①맨 마지막 칸을 제거한다.

②삭제된 위치의 다음 데이터를 삭제한 위치로 이동시킨다.

③마지막 위치까지 ②를 반복한다.

④삭제할 위치의 데이터를 이동시킨다.

4.다음은 선형 리스트의 맨 마지막에 빈칸을 추가하는 코드다. 빈칸을 채우시오.

보기 :

katok = ["다현", "정연", "쯔위", "사나", "지효", "모모"]

katok.append(None)

print(katok)

실행결과 :

['다현', '정연', '쯔위', '사나', '지효', '모모', None]

5. 다음은 선형 리스트의 맨 마지막 빈칸을 삭제하는 코드다. 빈칸을 채우시오.

katok = ['다현', '정연', '쯔위', '사나', '지효', '모모', None]

del(katok[6])

print(katok)

6. 각 행을 순서대로 조합해서 선형리스트에 데이터를 추가하는 함수를 만드시오

①katok[kLen-1] = friend

②katok.append(None)

③kLen = len(katok)

④def add\_data(friend) :

정답 : ④ ② ③ ①

```
def add_data(friend) :  
    katok.append(None)  
    kLen = len(katok)  
    katok[kLen-1] = friend
```

7. 다음은 선형리스트에 데이터를 추가하는 함수다. 빈칸에 들어갈 내용은?

보기 :

def insert\_data(position, friend) :

katok.append(None)

kLen=len(katok)

for i in range(kLen-1, position, -1)

katok[i] = katok[i-1]

katok[i-1] = None

katok[position] = friend

8. 다음 중 선형 리스트로 구현할 수 있는 응용 분야는?(정답 : ④)

① 트리 구조

② 그래프 구조

③ 최단 경로

④ 다항식

4장

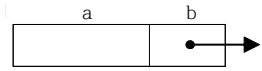
1. 괄호 안에 알맞은 용어를 각각 채우시오.

- (선형 리스트)는 배열에 데이터를 차례대로 저장하므로 데이터의 실제 위치 순서로 데이터가 구성된다.
- (단순연결리스트)에서는 데이터를 노드 단위로 삽입/삭제한다.

2. 선형리스트와 비교한 단순연결리스트에 대한 설명이다. 거리가 먼 것은?

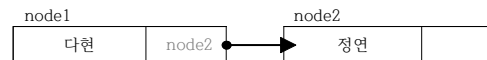
- ① 물리적으로 연결되어 있지 않은 데이터가 연결된다.
- ② 연결을 위한 링크가 필요하다.
- ③ 중간에 새로운 데이터를 삽입할 때는 비효율적이다.
- ④ 노드를 사용해서 데이터를 표현한다.

3. 노드 구조에서 a와 b를 무엇이라고 하는가?



정답 : a = 데이터, b = 링크

4. 그림과 같이 노드를 생성하고 연결하는 코드를 올바른 순서대로 나열하시오.



- a. node1.data= "다현"
- b. node1.link= node2
- c. node1 = Node()
- d. node2.data = "정연"
- e. node2 = Node()

정답 : c → a → e → d → b

```
node1 = Node()
node1.data= "다현"
node2 = Node()
node2.data = "정연"
node1.link= node2
```

5. 단순 연결 리스트의 맨 앞 데이터를 삭제하는 코드를 차례대로 나열하시오.

- a. head = head.link
- b. current = head
- c. del(current)

정답 : b → a → c

```
current = head
head = head.link
del(current)
```

6. 다음과 같이 코드가 작성되어있을 때, 쓰워를 출력하는 코드로 올바른 것은?

보기

```
node1 = Node(); node1.data="다현"
node2 = Node(); node2.data="정연"; node1.link = node2
node3 = Node(); node3.data="쓰워"; node2.link = node3
```

- ① print(node1)
- ② print(node1.data)
- ③ print(node1.link.data)
- ④ print(node1.link.link.data)

7. 단순 연결 리스트의 맨 앞에 야옹이 데이터를 삽입하려고 한다.

각 빈칸을 알맞은 코드로 채우시오.

```
node = Node()
node.data = "야옹이"
node.link = head
head = node
```

8. 노드에서 데이터를 검색하는 함수다. 빈칸에 들어갈 코드는?(답 : ②)

```
def findNode(findData) :
    global memory, head, current, pre
    current = head
    if current.data == findData :
        return current
    while current.link != None :
        current = current.link
```

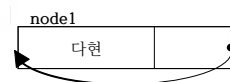
- ① while current.link == None
- ② while current.link != None
- ③ while head.link != None
- ④ while head.link == None

5장

1. 원형 연결 리스트의 특징과 거리가 먼 것 두가지를 고르시오.(정답 : ③,④)

- ① 단순 연결 리스트와 유사하지만 처음과 끝이 이어져있다.
- ② 노드 구조다.
- ③ 헤드는 필요 없다.
- ④ 마지막 노드의 링크는 비어있다.

2. 다음과 같은 원형 연결 리스트를 만드는 코드의 빈칸을 채우시오.



```
node1 = Node()
node1.data = "다현"
node1.link = node1
```

3. 지효에서 다현으로 이어지는 원형 연결리스트에서 쓰워 노드를 삭제하려 한다. 정연노드가 node2, 쓰워노드가 node3일 때, 빈칸을 알맞은 노드명으로 채우시오

다현 → 정연 → 쓰워 → 사나 → 지효 ...

정답 :

```
node3.link = node2.link
del(node3)
```

4. 원형 연결 리스트의 마지막 노드가 참이 되는 조건은?(정답 : ②)

- ① current.link != head
- ② current.link == head
- ③ current.link != None
- ④ current.link == None

5. 원형 연결 리스트의 첫 번째 노드를 삭제하려한다. 빈칸을 채우시오.

```
current = head
head = head.link
last = head
while last.link != current :
    last = last.link
last.link = head
del(current)
```

6. 원형 연결 리스트의 노드를 검색하는 함수다. 빈칸을 채우시오.

```
def findNode(findData) :
    global memory, head, current, pre
    current = head
    if current.data == findData :
        return current
    while current.data != head :
        current = current.link
    if current.data == findData :
        return current
    return Node()
```