

2025-2학기 알고리즘 중간고사

학과	학번	이름

※ 주의사항

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정합니다.
- 출력 예시에서 맨 앞에 □는 띄어쓰기를 의미합니다.
- 모든 정답 코드는 OJ시스템에 입력합니다.
- OJ시스템의 testdata에 문제에 제시된 입출력 예시가 포함되지 않은 경우도 있습니다.
- 문제에서 제시한 조건에 맞지 않게 구현한 경우 감점이 있습니다.

1. [20점] Padovan 수열 구현하기

Padovan 수열은 다음과 같이 정의된다. 소문제(1~2)를 풀어 제출하시오.

$$T_0 = T_1 = T_2 = 1$$

$$T_n = T_{n-2} + T_{n-3}, n \geq 3$$

1) [10점] 재귀를 이용한 방법

- 입력: n 의 값을 입력받는다. ($3 < n < 20$ 인 자연수)
- 출력: T_n 의 값을 출력하고, 재귀함수를 호출한 횟수를 출력한다.

입력 예시	출력 예시
10 → n	12 23 → T_n , 함수 호출 횟수

2) [10점] 비재귀를 이용한 방법

- 입력: n 의 값을 입력받는다. ($3 < n < 200$ 인 자연수)
- 출력: T_n 의 값을 출력한다.

입력 예시	출력 예시
7 → n	5 → T_n

2. [20점] 사전 이분탐색 구현하기

정렬되어 있는 N 개의 자연수 키(사전)와 탐색할 키 K 를 입력받아, 탐색경로와 K 의 위치를 출력하는 프로그램을 작성하시오.

◦ 구현 조건

- 크기가 N인 배열을 동적 할당하여, 입력된 사전의 키 저장(중복된 키는 없다고 가정)
- 이분탐색을 이용하여 탐색 키의 위치 찾기
- 배열의 인덱스는 0부터 시작한다.

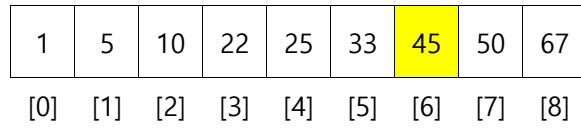
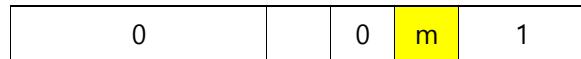
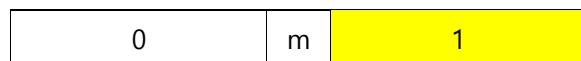
◦ 입력

- N의 값과 찾을 키 값 K를 입력받는다. (N은 2보다 큰 자연수)
- 오름차순으로 정렬된 N개의 자연수 키 값들을 입력받는다.

◦ 출력

- 탐색 경로를 출력한다. ($\lfloor N/2 \rfloor : 0, 2\lfloor N/2 \rfloor : 1, \dots$)

ex) N = 9, K = 45



- 키 K가 존재하는 경우에는 K의 위치를 출력하면 되고, 찾는 키가 없으면 -1 출력
- 중간값(mid)에서 키를 찾으면, 탐색을 중단하고 키 위치를 출력한다. (예시1,2 참고)

입력 예시1	출력 예시1
9 45 1 5 10 22 25 33 45 50 67	$\mapsto N, K$ \mapsto 키 값
입력 예시2	출력 예시2
9 25 1 5 10 22 25 33 45 50 67	4 \mapsto 탐색 경로, 25의 위치 (앞에 공백있음)
입력 예시3	출력 예시3
9 49 1 5 10 22 25 33 45 50 67	110 -1 \mapsto 탐색 경로, 49의 위치

3. [30점] 거리 기반 온도 추정하기

나의 **현재 위치(원점)**의 온도를 기상관측소에서 측정한 온도값을 토대로 추정하려고 한다.

다음 조건을 잘 읽고 소문제(1~3)를 풀어서 제출하시오.

- 구현 조건

- $n(n \leq 10)$ 개의 기상관측소의 위치좌표와 온도가 아래 예시와 같이 주어진다.
- 관측소의 위치 좌표는 정수로 주어지며, 거리가 동일한 관측소도 존재할 수 있다.
- i번째 기상관측소에서 측정한 온도를 T_i ,
- i번째 기상관측소까지의 거리를 d_i 라고 할 때,
- 현재 위치의 온도 T 는 다음 식으로 추정한다.

$$T = \frac{\sum_{i=1}^n \frac{T_i}{d_i^2}}{\sum_{i=1}^n \frac{1}{d_i^2}} = \frac{\frac{T_1}{d_1^2} + \frac{T_2}{d_2^2} + \frac{T_3}{d_3^2} + \cdots + \frac{T_n}{d_n^2}}{\frac{1}{d_1^2} + \frac{1}{d_2^2} + \frac{1}{d_3^2} + \cdots + \frac{1}{d_n^2}}$$

(각 관측소의 온도에 거리 제곱의 역수를 가중치로 부여하여, 가중 평균을 계산한다)

- 단, 현재 위치(원점)에 기상관측소가 존재하면, 그 기상관측소의 온도 측정값을 현재 위치의 온도 추정값으로 한다.

- 예시 ($n=5$ 일때)

- 기상관측소의 정보가 아래와 같이 제공되었다고 하자.

관측소이름	관측소의 위치	관측소에서 측정한 온도
A	(1, 2)	18
B	(3, 0)	23
C	(0, 3)	20
D	(-4, 2)	21
E	(3, -1)	19

- 이 때, 현재 위치(원점)에서의 온도 T 를 추정하는 과정은 다음과 같다.

$$\text{온도추정값 } T = \frac{\frac{18}{9} + \frac{23}{9} + \frac{20}{9} + \frac{21}{20} + \frac{19}{10}}{\frac{1}{9} + \frac{1}{9} + \frac{1}{9} + \frac{1}{20} + \frac{1}{10}} = \frac{2039}{103} = 19.7961165..$$

- 하지만, 만약 관측소 A의 위치가 원점(0, 0) 이었다면, 온도추정값은 18이어야 한다.

1) [10점] 구조체 배열 구현하기

아래와 같은 Station 구조체를 생성하고, 구조체 배열을 만들어 순서대로 출력하시오.

- 구조체 정보

```
typedef struct
{
    char name;
    int dist_sq;
    int temp;
} Station;
```

- name : 관측소의 이름
- dist_sq : 현재 위치로부터 관측소까지의 거리의 제곱
- temp : 관측소에서 측정한 온도

◦ 입출력 정보

입력 예시	출력 예시
5 A 1 2 18 B 3 0 23 C 0 3 20 D -4 2 21 E 3 -1 19	→ n A 5 18 B 9 23 C 9 20 D 20 21 E 10 19

2) [10점] dist_sq 기반 오름차순 정렬

이 구조체 배열을 dist_sq에 대한 오름차순으로 정렬하여 출력하시오.

◦ 구현 조건

- 거리가 동일한 경우, 입력된 순서대로 출력한다 (불안정한 정렬로 구현시 점수없음)

◦ 입출력 정보

입력 예시	출력 예시
5 A 1 2 18 B 3 0 23 C 0 3 20 D -4 2 21 E 3 -1 19	→ n A 5 18 B 9 23 C 9 20 E 10 19 D 20 21

3) [10점] 현재 위치의 온도 추정하기

앞서 제시한 수식에 따라 n개의 관측소 값을 모두 반영 계산하려 하니,

계산량이 너무 많아 현재 위치와 가까운 k개의 관측소만 이용해 온도를 추정하려 한다.

거리가 가까운 k개의 관측소의 온도만 고려하여, 현재위치의 온도를 소수점 셋째자리까지 추정하시오.

◦ 구현 조건

- 소문제 1,2와 달리 n, k가 입력으로 주어진다.
- 현재 위치에 관측소가 존재하면, 해당 관측소의 온도 측정값을 추정값으로 한다.
- 거리가 동일한 경우, 입력된 순서가 빠른 것을 우선선택한다.

(2번의 정렬상태를 그대로 이용하면 된다)

- 참고:

- int형끼리의 연산 결과는 정수로 계산된다.
실수 결과가 필요할 경우, 반드시 한쪽을 실수형으로 형변환해야 한다.
- 소수점 셋째자리까지 출력하는 법 `printf("%.3f", 값);`

◦ 입출력 정보

n=5, k=2일 때, 현재 위치(원점)에서의 온도 T를 추정하는 과정은 다음과 같다.

$$T = \frac{\frac{1}{5} \times 18 + \frac{1}{9} \times 23}{\frac{1}{5} + \frac{1}{9}} = \frac{277}{14} = 19.78571429..$$

입력 예시	출력 예시
5 2 A 1 2 18	→ n, k 19.786
B 3 0 23	
C 0 3 20	
D -4 2 21	
E 3 -1 19	

4. [30점] 스케줄러를 최소힙으로 구현하기

◦ 참고자료

운영체제는 현재 “준비 상태”(ready state)에 있는 여러 프로세스 중에서 어떤 프로세스에 CPU를 할당할지 결정한다. (준비상태란, 프로세스가 작업을 다 완료하고 실행할 준비를 마쳤지만, 아직 CPU를 할당받지 못해 기다리고 있는 상태를 말한다)

준비 상태에 있는 프로세스들은 운영체제의 “레디큐”에 들어가 CPU가 자신을 선택해 실행되길 기다린다.

레디큐에 있는 프로세스 중에 어떤 프로세스를 먼저 실행할지 결정하는 운영체제의 구성요소를 “스케줄러”라고 부른다.

운영체제는 프로세스에게 우선순위를 부여해놓고,

스케줄러는 레디큐에 있는 프로세스 중 우선순위가 높은 프로세스를 골라 먼저 실행시킨다.

예를 들어 프로세스 A,B,C,D는 작업을 모두 마치고 준비상태에 있고,

E는 아직 작업을 수행 중이라 준비상태가 아니라면,

레디큐에는 A,B,C,D가 들어가 있을 것이다.

이때 CPU가 현재 작업을 마쳤다면, 스케줄러는 A,B,C 중 우선순위가 가장 높은 프로세스 하나를 선택해 CPU를 할당해준다.



이 문제에서는 이러한 스케줄러의 동작을 단순화하여,
최소힙을 이용한 우선순위큐 방식으로 스케줄러를 간단히 구현해보고자 한다.

다음 조건을 잘 읽고 소문제(1~3)를 풀어서 제출하시오.

◦ 구현 조건

- 레디큐는 아래와 같은 **Process 구조체**의 배열로 구현한다.

```
typedef struct
{
    char name;
    int priority;
} Process;
```

- name : 프로세스의 이름
- priority : 프로세스의 우선순위(작을수록 우선순위가 높다)
- 레디큐(구조체배열)의 크기는 10이다.
- 우선순위가 동일한 프로세스는 존재하지 않는다고 가정한다.
- 레디큐는 오름차순 완전정렬상태가 아닌, 최소힙 상태를 유지함에 주의하라.
- 최소힙은 key(parent) <= key(child)를 만족하는 힙으로 정의한다. 불필요한 swap이 발생시 감점될 수 있음.

1) [10점] 최소힙을 이용해 레디큐 만들기

다음 조건을 따라 레디큐를 만들고 완성된 레디큐의 상태를 출력하시오.

◦ 구현 조건

- 삽입식으로 힙을 생성하시오.
(상향식으로 생성해도 이 문항에서의 감점은 없으나, 소문제 2번은 삽입식으로 구현해야 풀 수 있다)
 - priority 값을 키로 갖는 최소힙 상태를 유지하시오.
 - 구현의 편의성을 위해 배열의 0번 인덱스는 비워두시오.
 - upHeap()을 이용해 구현하시오.
 - 아래와 같이 명령어를 정의한다. (d는 소문제 2번에서만 쓰인다)
 - i : 레디큐에 프로세스를 삽입한다.
 - d : CPU 할당을 완료하였기에 프로세스를 레디큐에서 삭제한다.
 - q : 프로그램을 종료한다.
 - i, d, q외의 입력은 제공되지 않는다.
 - 명령은 9개 이하로만 제시된다. (원형큐로 구현할 필요 없다)
- 입출력정보

- 입력: (명령어 i/d/q)-(프로세스의 이름)-(프로세스의 우선순위)가 순서대로 입력된다.
- 출력: 레디큐 구조체 배열을 출력한다.

입력 예시 (\rightarrow 레디큐의 상태)	출력 예시
i A 3	\rightarrow A
i B 7	\rightarrow AB
i C 5	\rightarrow ABC
i D 1	\rightarrow DACB
i E 0	\rightarrow EDCBA
q	

- 입력 받은 후 레디큐의 상태는 아래와 같아야 한다.

인덱스	0	1	2	3	4	5	6	7	8	9
name		E	D	C	B	A				
priority		0	1	5	7	3				

2) [20점] 스케줄러 구현하기

주어진 입력에 따라 차례로 레디큐에 삽입과 삭제가 일어날 때, 최종 레디큐의 상태를 출력하시오.

- 참고자료

스케줄러가 레디큐에 있는 프로세스 중 하나에 CPU를 할당하면, 해당 프로세스는 더 이상 대기할 필요가 없으므로 레디큐에서 제거되어야 한다.

이때, 삭제 후에도 최소힙 구조를 유지하여, 다음 스케줄링 시에도 우선순위가 가장 높은 프로세스를 즉시 선택할 수 있도록 해야한다.

- 구현 조건

- 삽입 직후, 삭제 직후 모두 최소힙 상태를 유지해야 한다.
- `upheap()`, `downHeap()`을 이용해 구현하시오.
- 프로세스큐가 비어있는 상태에서 프로세스가 삭제되는 경우는 없다고 가정한다.
- 명령은 9개 이하로만 제시된다. (원형큐로 구현할 필요는 없다)

- 프로세스 삭제 예시

- 스케줄러가 우선순위가 가장 높은 프로세스 E를 삭제한 직후의 레디큐의 상태는 아래와 같아야 한다.

인덱스	0	1	2	3	4	5	6	7	8	9
name		D	A	C	B					

priority		1	3	5	7					
----------	--	---	---	---	---	--	--	--	--	--

◦ 입출력 정보

입력 예시 (\rightarrow 레디큐의 상태)	출력 예시
i A 3	\rightarrow A
i B 7	\rightarrow AB
i C 5	\rightarrow ABC
i D 1	\rightarrow DACB
i E 0	\rightarrow EDCBA
d	\rightarrow DACB
d	\rightarrow ABC
i F 2	\rightarrow FACB
i G 4	\rightarrow FACBG
d	\rightarrow AGCB
q	