

CSE3013 컴퓨터 공학 설계 및 실험I (Spring 2020)

기말고사

2020.06.23 (TUE)

시험시간

- 19:00 - 19:30 본인확인, 개인별 준비 및 문제 배포.
- 19:30 - 21:30 실기 시험.
- 21:30 시험 종료 및 서버 로그아웃

시험 문제는 아래와 같이 총 5개의 문제가 주어진다. 학생들이 해결해야하는 문제의 수는 총 4개이다. 학생들은 3개의 필수 문제를 해결하고 선택 문제 2개의 문제 중 하나의 문제를 선택해서 풀면 된다. (총 4문제만 풀면 됩니다.)

시험 총점은 100점으로 환산 된다. 각 문제를 주의 깊게 읽어 보고 문제를 충분히 이해한 후 선택 문제 2개중 하나의 문제를 선택해서 프로그래밍을 시작하기를 권고한다.

- 문제1. 신입 사원 채용(필수)
- 문제2. 랭킹 시스템 (필수)
- 문제3. 요세푸스 문제 (필수)
- 문제4. 점프 (선택)
- 문제5. 오렌지/사과 예쁘게 진열하기 (선택)

시험 시간 주의사항

- 시험 중 타인의 자료를 볼 수 없으며, 네트워크(인터넷)도 사용할 수 없습니다.
- 어떠한 대화도 불허합니다.
- 화장실은 감독 조교의 허락을 받아야 합니다.
- 실험 교재, C언어(1권), 자료구조 교재(1권)만 열람할 수 있습니다.
- 듀얼 모니터는 사용 불가입니다.
- 노트북과 데스크탑을 연결하여 사용하신다면 하나의 화면만 켜져 있어야 합니다.
- 태블릿 등 다른 전자기기로 참고자료를 열람하는 것은 불허합니다.
- 카카오톡, 네이버온 같은 메신저가 켜져 있으면 부정행위 입니다. 반드시 시험 전 각종 메신저를 종료하시기 바랍니다.
- 시험을 조기 종료하는 것은 허용되지 않습니다. 시험이 종료될때까지 자리를 지켜주시기 바랍니다.

프로그램 제출 안내

- 홈 디렉토리(접속하면 뜨는 화면이 홈 디렉토리입니다.)에 s_학번으로 된 디렉토리를 생성합니다. (ex. s_20190001) 이때 s는 소문자입니다.
- 디렉토리를 생성하는 명령어는 mkdir입니다. s_20190001의 이름을 가진 디렉토리를 생성하고자 하면 mkdir s_20190001을 입력하시면 됩니다.
- 본인의 학번으로 생성된 디렉토리 내에서 파일을 생성하여 문제를 해결합니다.
- 파일이름: **xxxxxxxx_n.cpp** (xxxxxxxx : 학번 8자리, n : 문제 번호(1~ 5)). 'cpp'는 소문자.
- Ex) 학번이 20190001인 학생이 3번 문제를 푼다면 20190001_3.cpp 와 같은 파일명을 생성 하셔야 합니다.

제출 방법 주의사항

1. 제출은 서버에서 일괄적으로 학생들의 디렉토리를 수거할 예정이기에 모든 학생은 시험 종료 (21:30)시간이 되면 cspro계정에서 로그아웃을 하여야 합니다.
2. 시험시간이 종료되면 일괄적으로 서버에서 학생들의 디렉토리를 수거해갈 것이기에 추가시간을 드릴 수 없으니 종료 10분 전부터는 파일을 저장하시면서 코딩을 하시는 것을 추천 드립니다.

문제 1. 신입 사원 채용 (필수)

회사 A의 사원 채용 과정은 서류심사, 면접시험 2 단계로 이루어지며 점수는 독립적으로 매겨진다. 회사 A에서는 최고의 인재들을 채용하고 싶기 때문에 한 가지 심사 기준을 세웠다.

A에서는 어떤 지원자의 두 가지 점수가 다른 어떤 지원자의 두 가지 점수와 비교하여 모두 떨어질 경우 채용하지 않기로 하였다. 즉 어떤 지원자 a의 서류 심사 성적, 면접 시험 성적 각각이 지원자 b의 그것보다 낮다면 a는 선발되지 않는다.

이러한 조건 하에서 회사 A가 이번 공채에서 선발할 수 있는 최대 인원수를 구하는 프로그램을 작성하여라.

입력

첫째 줄에는 테스트 케이스의 개수 $T(1 \leq T \leq 20)$ 가 주어진다. 각 테스트 케이스의 첫째 줄에 지원자의 숫자 $N(1 \leq N \leq 100,000)$ 이 주어진다. 둘째 줄부터 N 개 줄에는 각각의 지원자의 서류심사 성적, 면접 성적의 순위가 공백을 사이에 두고 한 줄에 주어진다. 두 성적 순위는 모두 1위부터 N 위까지 동석차 없이 결정된다고 가정한다.

출력

각 테스트 케이스에 대해서 회사 A가 선발할 수 있는 신입사원의 최대 인원수를 한 줄에 하나씩 출력한다.

입출력 예시

입력	출력
2 5 3 2 1 4 4 1 2 3 5 5 7 3 6 7 3 4 2 1 4 5 7 2 5 6 1	4 3

문제2 랭킹 시스템 (필수)

철수는 실습 때 테트리스 게임을 구현하였다. 철수는 이제 만들어진 테트리스 게임에 랭킹 시스템을 구현하고자 한다. 철수는 링크드리스트를 사용하여 입력으로 들어오는 사용자의 점수를 내림차순으로 정렬하여 출력해주려 한다. 철수는 동적 할당을 이용하여 링크드리스트를 구현할 것이기 때문에 몇 개의 입력이 들어오는지 알 필요 없이 해당 프로그램을 구현할 수 있다고 한다. 자신만만한 철수를 도와 랭킹 시스템을 구현해보자. (반드시 링크드 리스트를 사용하여 구현 하셔야 됩니다. 배열로 구현 시 오답입니다.)

입력

첫째 줄에는 테스트 케이스의 개수 $T(1 \leq T \leq 20)$ 가 주어진다.

둘째 줄부터 N 개 줄에는 정렬하고자 하는 점수가 한 칸의 공백을 두고 입력된다. -1이 입력될 시 해당 줄의 입력이 종료된다. -1은 입력되는 점수로 취급하지 않는다. 입력으로 들어오는 점수는 양의 정수이고, 동점은 없다고 가정한다. 입력으로 들어오는 점수의 개수는 최소 2개 이상 최대 100개이다.

출력

각 테스트 케이스에 대해서 내림차순으로 정렬된 점수가 한 줄씩 출력된다. 각 줄 안의 출력은 한 칸씩 띄어서 출력이 된다.

입출력 예시

입력	출력
3	5 4 3 2 1
3 2 1 4 5 -1	10 2
10 2 -1	1000 20 10
10 1000 20 -1	

문제3 요세푸스 (필수)

요세푸스 문제는 다음과 같다.

1번부터 N 번까지 N 명의 사람이 원을 이루면서 앉아있고, 양의 정수 $K(K \leq N)$ 가 주어진다. 이제 순서대로 K 번째 사람을 제거한다. 한 사람이 제거되면 남은 사람들로 이루어진 원을 따라 이 과정을 계속해 나간다. 이 과정은 N 명의 사람이 모두 제거될 때까지 계속된다. 원에서 사람들이 제거되는 순서를 (N, K) -요세푸스 순열이라고 한다. 예를 들어 $(7, 3)$ -요세푸스 순열은 $\langle 3, 6, 2, 7, 5, 1, 4 \rangle$ 이다. N 과 K 가 주어지면 (N, K) -요세푸스 순열을 구하는 프로그램을 작성하시오.

단, $(1 \leq K \leq N \leq 5,000)$

입출력 예시

입력	출력
7 3	<3, 6, 2, 7, 5, 1, 4>

문제 4. 점프(선택)

$N \times N$ 게임판에 수가 적혀져 있다. 이 게임의 목표는 가장 왼쪽 위 칸에서 가장 오른쪽 아래 칸으로 규칙에 맞게 점프를 해서 가는 것이다.

각 칸에 적혀있는 수는 현재 칸에서 갈 수 있는 거리를 의미한다. 반드시 오른쪽이나 아래쪽으로만 이동해야 한다. 0은 더 이상 진행을 막는 종착점이며, 항상 현재 칸에 적혀있는 수만큼 오른쪽이나 아래로 가야 한다. 한 번 점프를 할 때, 방향을 바꾸면 안 된다. 즉, 한 칸에서 오른쪽으로 점프를 하거나, 아래로 점프를 하는 두 경우만 존재한다.

가장 왼쪽 위 칸에서 가장 오른쪽 아래 칸으로 규칙에 맞게 이동할 수 있는 경로의 개수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 게임 판의 크기 N ($4 \leq N \leq 100$)이 주어진다. 그 다음 N 개 줄에는 각 칸에 적혀져 있는 수가 N 개씩 주어진다. 칸에 적혀있는 수는 0보다 크거나 같고, 9보다 작거나 같은 정수이며, 가장 오른쪽 아래 칸에는 항상 0이 주어진다.

출력

가장 왼쪽 위 칸에서 가장 오른쪽 아래 칸으로 문제의 규칙에 맞게 갈 수 있는 경로의 개수를 출력한다. 경로의 개수는 $2^{63}-1$ 보다 작거나 같다.

입출력 예시

입력

```
4
2 3 3 1
1 2 1 3
1 2 3 1
3 1 1 0
```

출력

```
3
```

문제 5. 오렌지/사과 예쁘게 진열하기 (선택)

당신은 마트에서 상품 정리를 하고있다. 현재 아래 그림과 같이 오렌지와 사과가 한 줄에 무작위로 진열되어 있다.

그림 1. 오렌지와 사과가 무작위로 진열되어 있는 모습

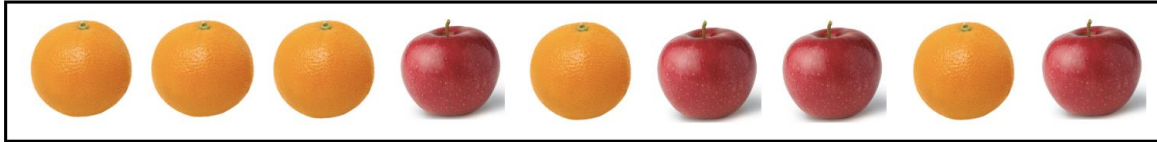


그림 2. 예쁘게 진열된 오렌지와 사과의 모습



당신은 무작위로 배열된 상품을 예쁘게 진열하기 위해서 오렌지와 사과를 같은 상품은 인접하도록 일렬로 진열하고 싶다. 당신은 한 번에 하나의 상품을 다른 상품으로 바꿀 수 있다. 이를 작업 K 라고 하겠다. 예를 들어 오렌지를 사과로 바꾸거나 사과를 오렌지로 바꿔서 진열 할 수 있다.

당신은 이 작업 K 가 매우 귀찮기 때문에 가장 최소한의 노력으로 상품을 예쁘게 진열하고 싶다. 꼭 두개의 상품이 모두 진열 될 필요는 없으며 하나의 상품만 진열되어도 상관 없다. 오렌지와 사과의 진열 목록이 배열 ARR 가 주어졌을 때 K 작업을 가장 적게 수행하여 상품을 예쁘게 진열할 수 있는 방법을 찾아내고 수행한 K 작업의 횟수를 출력하시오. (단, 첫 번째 상품은 반드시 오렌지 이어야 한다)

진열된 상품의 정보는 다음과 같이 문자열로 주어진다.

O = 오렌지

A = 사과

진열된 상품 정보 = OOOAOAAOA

입력 첫 줄에는 총 테스트 케이스 $T(1 \leq T \leq 10^4)$ 를 입력받는다. 한 테스트 케이스에 대해서는 두개의 줄에 걸쳐 입력 받는다. 케이스 T_i 의 첫 번째 줄에는 진열된 상품의 개수 $N(1 \leq N \leq 10^6)$ 을 입력 받고 그 다음 줄에는 상품의 목록을 문자열로 입력받는다.

출력 예쁘게 상품을 진열하기 위해 가장 적게 상품을 바꾸어야 하는 횟수를 출력한다.

입출력 예시

입력	출력
4	0
3	1
OOA	1
5	2
OOAOA	
1	
OOOA0000	
4	
AOAO	

설명

첫 번째 케이스인 OOA 는 이미 잘 진열되어 있기 때문에 상품을 바꾸지 않아도 된다.

두 번째 케이스인 OOAOA 는 4 번째 오렌지를 사과로 바꾸면 OOAAA 로 예쁘게 진열되기 때문에 출력 값은 1 이 된다.

세 번째 케이스인 OOOA00000 는 진열된 하나의 사과를 오렌지로 바꾸면 OOOO000000 가 되기 때문에 출력 값은 1 이 된다.

네 번째 케이스인 AOAO 에서는 첫 번째 상품이 오렌지 이어야 하므로 사과를 오렌지로 바꾸어 OOAO 로 만들고 여기서 OOOO 로 진열하거나 혹은 OOAA 로 진열하는 두 가지 방법이 있다. 두 방법 모두 상품을 바꾸는 행위를 2 번만 하므로 출력 값은 2 가 된다.