



연구원 모집

[문제] Super-AI 연구소 삼돌이(SAMDORI)¹⁾에서는 필요한 시점에 수시로 재능있는 연구원을 모집한다. 여기에 취업을 희망하는 졸업자들은 이 연구소의 서버에 지원서를 등록하고 선발을 기다린다. 그리고 지원자들이 일정 수 이상 모이면, 연구소에서는 이 등록된 지원자들을 대상으로 그 우선 순위를 매긴 다음에 그 중 일부를 연구원으로 선발한다.

지원자들이 제공(등록)하는 정보는 3개로 구성되어 있다. 먼저 지원자의 고유한 ID, 예를 들어 주민등록번호 같은 것이다. 이 ID는 6자리 정수로 표시되며 개인별로 유일(unique)하다. 그 다음에는 자신이 졸업한 대학과 학과를 표시한 졸업(GRAD) 코드(code)이다. 이 코드는 영문 대문자 7자로 표시된다. 예를 들어 PNUCSED, PNUAIDP, YSDCVDP, KORCSED²⁾ 등이다. 마지막 정보는 이 연구소에서 실시한 코딩 테스트(COTE) 최종 평가점수로 $[0, 1000]$ 사이의 정수 값이 주어진다. SAMDO 연구소 지원자들의 수시 모집 과정은 다음과 같다.

1. 일단 지원자들은 같은 졸업 코드 GRAD 단위로 분류한다.
2. 같은 졸업 코드를 가진 지원자들은 그 안에서 COTE 높은 순으로 정렬된다.
3. 각 GRAD의 순서는 지원자가 많은 순으로 정렬되며, 즉 지원자가 많은 대학이 더 우선되며
4. 지원자의 수가 같을 경우에는 COTE 점수의 합이 높은 쪽 대학이 우선된다.
5. GRAD별로 지원자가 일정 수 k 이상인 대학만 고려대상이 된다. 이 상수 k 는 모집 때마다 각각 지정된다. 예를 들어 $k=2$ 라면 지원자가 1명 이하인 대학은 고려되지 않는다. 이런 k 명 이상의 지원자가 있는 GRAD 대학/학과를 선발가능대학(CANU)이라고 부른다. CANU는 지원자의 수에 따라서 변한다. 만일 지원자 중 일부가 선발되어 대기 DB에서 빠져나가 k 명 이하로 되면 이 대학은 k 명으로 채워질 때까지 고려대상이 되지 않는다.
6. k 명 이상 지원한 대학학과, 즉 선발가능대학(CANU)의 1위(COTE 기준) 지원자들을 대학 순위대로 순위를 매긴다. 즉 BANANAD 지원자가 8명, APPLEDE 지원자가 6명이라면 BANANAD 대학의 1위가 COTE 점수가 APPLEDE 1위보다 더 낮아도 더 높은 순위를 가진다.
7. 그 다음 순위는 선발가능대학의 각 2위들을 뽑아서 같은 식으로 순위를 매긴다. 이 과정에서 해당 순위의 지원자가 없으면 무시한다.
8. 위 7번 과정을 해당 순위자에 도달할 때까지 반복한다.

예를 들어 설명해보자. 현재 지원자의 상태를 표로 만들어 보면 다음과 같다. 아래 각 칸에서 위에 있는 숫자는 지원자의 ID이며 그 아래 숫자는 해당 지원자의 COTE 점수이다. PNUCSED에는 이미 6명이 지원한 상태이며 그들은 COTE 점수 순서로 정렬되어 있다. 만일 $k=3$ 인 경우라면

1) Super AI Manage & Development Organizing Research Institute

2) 각각은 부산대학교 컴공, 부산대학교 AI전공, 연세대학교 산업공학과, 고려대학교 컴공을 나타낸다.

KAISTCS와 DUMMYUM는 선발 후보에서 모두 제외된다. 즉 우리는 DB에 3명 이상인 대학만 선발에 고려한다.

GRAD Code	1위	2위	3위	4위	5위	6위	7위
PNUCSED	234561 878	457232 778	112453 750	892826 683	111780 645	378904 590	
KAISTCS	901929 756	872728 567					
KOREACS	562436 856	272727 701	837272 678	112435 657	563872 578	902534 545	448878 526
SOGANGD	900014 768	820202 733	700149 686				
PNUCVED	511234 766	340070 686	619034 633				
POSTECD	901234 806	173600 786	101410 733	666888 689	700034 656		
DUMMYUM	440055 930						

위 표에서 Green으로 표시된 대학만 “현재” 선발 후보가 된다. 이 상황에서 순위를 매겨보자. 먼저 지원자 수에 따른 대학 순위, 같을 경우 COTE 점수의 합으로 정렬하면 아래 표와 같아진다.

GRAD Code	1위	2위	3위	4위	5위	6위	7위
KOREACS	562436 856	272727 701	837272 678	112435 657	563872 578	902534 545	448878 526
PNUCSED	234561 878	457232 778	112453 750	892826 683	111780 645	378904 590	
POSTECD	901234 806	173600 786	101410 733	666888 689	700034 656		
SOGANGD	900014 768	820202 733	700149 686				
PNUCVED	511234 766	340070 686	619034 633				

등위는 각 대학별 1위부터 순서대로 “걸어”내면 된다. 즉 그 순서는 562436, 234561, 901234, 900014, 511234, 272727, 457232, 173600, 820202, 340070, 837272,... 902534, 378904, 448878 순이다.3) 따라서 만일 5등에서 7등까지만 뽑는다면 그 선발된 지원자는 **[511234, 272727, 457232]**가 된다. 1등부터 뽑지 않는 이유는 최상위 지원자들의 공통적인 특성 때문이다.4)

3) 국제 대학생 프로그래밍 대회(ICPC)에서도 이와 유사하게 선발한다. 이렇게 선발하는 이유는 특정 대학만 뽑을 경우 내부에 여러 문제가 생길 수 있기 때문이다. 해외라면 인종, 국가 등도 고려해서 최대한 우수한 사람은 다양하게 선발한다.

4) 하바드 교수 뽑기 문제라고도 하는데, 최상위 지원자는 이미 다른 여러 곳에도 지원을 하여 합격할 가능성이 매우 높기 때문에 이들을 선발하면 대부분 최종 지원을 하지 않기 때문이다. 또 최상위 지원자는 취업을 어려움을 잘 모르기 때문에 회사에 대한 충성심이 약하여 언제든지 “도망”갈 생각을 하기 때문이다- 라고 한다.

지원자들은 수시로 지원하며 업체에서는 일정 시간이 흐른 뒤에 필요한 인원만큼 원하는 순위 범위에 있는 지원자들을 골라 선발한다. 이 상황에서 새로운 GRAD 코드와 새로운 지원자가 들어온다면 다시 그 자리를 찾아서 insert 해야 한다. 위 상황에서 [5, 7] 등 3명의 지원자를 선발한 뒤의 상황은 다음과 같이 변할 것이다. 즉 선발된 인원들은 해당 대기 자료구조에서 삭제되어야 한다. 만일 남아있는 사람의 수보다 선발해야 할 사람이 많을 경우에는 그 순위까지만 선발한다. 즉 선발 가능한 후보자가 9명일 경우 선발 순위가 [7,11]로 주어지면 10위, 11위는 존재하지 않기 때문에 7,8,9위, 이 3 사람만 선발해야 한다.

GRAD Code	1위	2위	3위	4위	5위	6위	7위
KOREACS	562436 856	837272 678	112435 657	563872 578	902534 545	448878 526	
PNUCSED	234561 878	112453 750	892826 683	111780 645	378904 590		
POSTECD	901234 806	173600 786	101410 733	666888 689	700034 656		
SOGANGD	900014 768	820202 733	700149 686				
PNUCVED	340070 686	619034 633					

위 상황에서 “PNUCVED 300900 783”과 “MANGOCS 562330 344”, “PNUCSED 555888 600” 이 3명의 지원자가 추가 등록하면 대기자 DB는 다음과 같이 바뀐다.

GRAD Code	1위	2위	3위	4위	5위	6위	7위
KOREACS	562436 856	837272 678	112435 657	563872 578	902534 545	448878 526	
PNUCSED	234561 878	112453 750	892826 683	111780 645	555888 600	378904 590	
POSTECD	901234 806	173600 786	101410 733	666888 689	700034 656		
SOGANGD	900014 768	820202 733	700149 686				
PNUCVED	300900 783	340070 686	619034 633				
MANGOCS	562330 344						

지원자들이 수시로 지원하는 상황에서 연구소에서는 중간 중간에 필요한 수의 지원자를 현재의 지원자 DB에서 선발한다. 여러분은 이 과정에서 각 단계마다 뽑힌 지원자들이 누구인지 찾아서 출력해야 한다. .

[입출력] 표준 입출력을 사용한다. 첫 줄에는 N 과 k 가 주어진다. N 은 그 아래 제공되는 줄(line)의 수이며 각 줄은 지원자의 정보 혹은 POP(선발) 작업으로 구성되어 있다. k 는 최저 지원자 기준이다. 단 $5 \leq N \leq 1000$, $2 \leq k \leq 10$ 이다. 우리는 현재 지원자 DB 중에서 k 명 이상인 지원자 대학만을 고려한다.

이어지는 N 개의 줄에는 새로 등록한 지원자의 정보 혹은 POP 정보가 주어진다. 지원자의 경우 그 정보는 대학 코드, 지원자ID, COTE 점수, 3개가 문자열, 정수, 정수로 주어진다. 대학 코드는 영문 대문자 7자, UID는 6자리 정수로 모두 unique하다. 끝 정수는 $[0,1000]$ 범위의 COTE 성적을 나타낸다. 만일 그 줄의 첫 단어가 POP이며 선발 작업이 진행됨을 의미한다. POP 뒤에는 2개의 정수 p, q 가 주어지는데 이는 연구원으로 선발하는 등위의 시작과 끝을 의미한다. 단 $p \leq q$ 이다. $p=q$ 이면 해당 등위의 지원자 1인만 선발하는 것이다. 이때 p, q 의 범위가 k 로 1차적으로 걸어진 학생 수 보다 크면 출력하지 않는다. 선발된 지원자는 대기 DB에서 삭제된다. 여러분은 뽑힌 지원자를 한 줄에 등위 순서대로 출력해야 한다.

[예제 1]

stdin	stdout
45 3 //N=45, k=3	
SOGANGD 900014 768	511234 272727 457232 //pop 5 7
PNUCSED 112473 750	897565 //pop 4 4
POSTECD 700034 656	700321 802991 //pop 13 14
PNUCSED 892826 683	
....	
(omitted)	
...	
KOREACS 562436 856	
PNUCVED 340070 686	
KIASTCS 901929 756	
POP 5 7	
YONSECS 334455 403	
....	
....	
POP 4 4 // Get 4 th	
PNUCSED 172400 545	
POSTECD 645039 754	
....	
POP 13 14	
END // T	

[제한조건] 프로그램 이름은 **recruit.{c,cpp,py}**이다. 마감 기한과 크기 제한은 모두 NESPA를 참조하기 바란다. STL에서 제공하는 함수를 적극적으로 사용해야 한다. 간결하고 가장 우아한 코드 Top 3에게 가점이 있다. 단순히 Token 개수만이 아니라 적절히 짧으면서도 이해가 쉽고 STL 제공함수를 잘 활용한 코드를 기대한다.