



☆ DNA 정렬 출력

문제 설명

최근 발견 된 N 개의 생명체를 연구하기 위해 각 개체의 DNA 배열 S 와 적응력(A)이 저장되어 있는 극비 데이터가 있습니다.

생명체들에 대한 다양한 실험을 위해 적응력이 Q 이상인 생명체들의 DNA 배열을 모두 찾아서 개체수가 많은 순으로 최대 K 개의 DNA배열을 개체수와 함께 출력하는 프로그램을 작성하세요.

(단, 출력 할 결과가 K 개 미만일 경우 해당 결과만 출력하세요. 동일한 개체수를 갖는 DNA 배열이 여러 개일 경우 'C', 'G', 'A', 'T' 순으로 정렬 했을 때 앞에 있는 DNA가 우선 순위를 갖습니다. 예를 들어, "CGTT"는 "CAGT" 보다 우선 순위를 갖습니다.)

입력

첫번째 줄에는 데이터 개수를 나타내는 숫자 N ($1 \leq N \leq 10,000,000$)이 주어지고 그 다음 N 개의 줄에 각 생명체의 DNA 배열 S 와 적응력 A ($0 \leq A < 100$)가 공백으로 구분되어 있습니다.

이 문제에서 모든 생명체의 DNA 배열은 128개의 'A', 'C', 'G', 'T'로 구성됩니다.

$N+1$ 번째 줄에는 실험 횟수를 나타내는 숫자 Q 가 주어지고

```

60     }
61     return true;
62 }
63 }
64
65 void solve(int q, int k) {
66     vector<pair<int, int>> candidate;
67     for(int i=0; i<htable.size(); i++) {
68         int sum = accumulate(&htable[i].nrOrder[q],
69                             &htable[i].nrOrder[100], 0);
70         //      htable[i].nrOrder[q];
71         candidate.push_back(make_pair(sum, i));
72     }
73     sort(candidate.begin(), candidate.end(),
74          candidateComprator);
75     printf("%d\n", min(k, (int)candidate.size()));
76     for (int i=0; i<candidate.size() && i < k; i++) {
77         printf("%d %s\n", candidate[i].first,
78              htable[candidate[i].second].dna.c_str());
79     }
80 }
81 void insertDnaTable(int a, string dna) {
82     long long int
83     dnaNumber[DNA_LEN/DNA_NUMBER_DIVIDER];
84     dnaStringToNumber((char*)dna.c_str(), dnaNumber);
85     //      vector<struct dnaTable>
86     vector<struct dnaTable>::iterator pos;
87     pos = find(htable.begin(), htable.end(), dnaNumber);
88     if (pos == htable.end()) {
89         struct dnaTable d;
90         memcpy(d.dnaNumber, dnaNumber,
91              sizeof(d.dnaNumber));
92         memset(d.nrOrder, 0, sizeof(d.nrOrder));
93         d.dna = dna;
94         d.nrOrder[a] = 1;
95         htable.push_back(d);
96         //      printf("NEW THING\n");
97     }
98     else {
99         pos->nrOrder[a]++;
100        //      printf("F");
101    }

```