#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#pragma warning(disable: 4996)

#define MAX 20 //n이 작은수의 경우가 어느정도인지 정확히 기제되어있지않아서 나중에 고치기 편하게 MAX에 값 저장

void arraySwap(int \* L, int n1, int n2) {

int temp; // 임시로 값을 저장할 변수

temp = L[n1];

L[n1] = L[n2];

L[n2] = temp;

// A[n1]의 값과 A[n2]의 값을 swap한다

}

void downHeap(int i, int n, int \* L) {

int left, right, smaller;

left = i \* 2;

right = left + 1;

if (left > n)//외부노드일때

return ;

smaller = left;

if (right <= n) {

if (L[right] < L[smaller])//smaller의 위치에 있는 키값이 right의 위치에 있는 키값보다 작으면 /최소힙

smaller = right;

}

if (L[i] <= L[smaller]) // i의 위치에 있는 값이 smaller의 위치에 있는 값보다 작거나 같아야만 리턴 / 최소힙

return ;

arraySwap(L, i, smaller);

downHeap(smaller,n,L);//재귀함수

}

void buildHeap(int n,int \*L) {//힙생성

int i;

for (i = n / 2; i >= 1; i--)

downHeap(i,n,L);

}

int findKthSmallest(int \* L,int n, int k) {

int i, e;

int last = n;//리스트의 크기를 저장

buildHeap(n,L);

for (i = 0; i < k; i++) {

e = L[1];//최솟값 저장

arraySwap(L, 1, last);

last--;//리스트크기 하나 줄임

downHeap(1,last,L);

}

return e;//k순위 원소

}

int main() {

int n, i, \*L, k , e;

printf("리스트 크기:");

scanf("%d", &n);

L = (int \*)malloc(sizeof(int)\*(n + 1));

srand(time(NULL));

for (i = 1; i <= n; i++) {

\*(L + i) = (rand()\*rand()) % 1000000 + 1;// rand()는 범위가 0~32767까지만이다. 따라서 1~1000000사이의 정수를 발생시키기 위해 rand()\*rand()를 해서 범위를 넓히고 %1000000+1을 한다

}

if (n <= MAX) {

printf("리스트 :");

for (i = 1; i <= n; i++)

printf(" %d", L[i]);

printf("\n");

}

buildHeap(n, L);

for (i = 0; i < 3; i++) {

printf("순위: ");

scanf("%d", &k);

e = findKthSmallest(L,n, k);

printf("원소: %d\n", e);

}

}