COP 3502 Suggested Program Edits/Questions: Binary Search Applications (Week 14 Programs)

1) Rewrite either of the etch solutions using a for loop that runs 60 times instead of the while(high – low > EPSILON) loop.

2) Pick one of the functions in the solution to approach.c, erase it, and see if you can rewrite it. Good candidates for this exercise are: getMaxRec, getMaxTime, and works.

3) Use the binary search technique to write a function that calculates the cuberoot of a number. Your function can use the following function signature:

```
double cuberoot(double x);
```

4) Write a solution to the Airport Shuttle problem described below:

When all of the out of state campers arrived to Orlando International Airport, the SI@UCF staff had to make several runs to the airport to pick up all of the campers. Naturally, none of the staff members wanted to wait too long at the airport. Each staffer's wait time was simply the difference in arrival times between the first and last camper he/she picked up.

Luckily, all of the staffers have access to arbitrarily large shuttle buses! But, in addition to wanting to minimize their wait time at the airport, none of the staffers want to make more than one airport run.

Given the number of SI@UCF staffers, as well as the arrival times of each camper flying into Orlando International Airport, determine the minimum amount of time, T, in minutes, such that no staffer will have to wait more than T minutes.

Input
The first line of input contains two space separated positive integers: n ($n \leq 10^5$), and k ($k \leq n$), where n represents the number of campers flying into the airport and k represents the number of staffers. The second line of input contains the n arrival times, in minutes, separated by spaces, of each camper. Each of these arrival times will be positive integers less than or equal to 10^9.

Output
On a single line by itself, output the minimum number of minutes T, such that no staffer waits more than T minutes, no staffer takes more than one trip, and each camper gets picked up.

Samples

Input	Output
9 3 10 10 30 200 205 210 215 220 500	20
5 2 10 2 16 19 5	8