Little Deepu and his Girlfriend - Hackerearth

Sunday, February 23, 2020 12:46 PM

The problem:

To put the game a bit formally, we have two players Little Deepu and Kate and M items in the bag B, also we have a game set S with N elements, each element of game set is an integer. The game is played as follows, each player takes turn to pick an element from the game set S and removes that number of items from the bag, the player that is unable to remove the items from the bag looses the game. Little Deepu start the game ,If both Little Deepu and Kate play the game optimally, your task is to determine who wins the game.

Input:
First line contains a integer T , number of test cases. Each test case contain two lines , first line contain two integer M and N and second line contain elements of S.

Output:
For each test case print name of the winner of the game .
Constraints:
$1 \leq T \leq 1000$
$1 \leq M \leq 10000$
$1 \leq N \leq 100$
$1 \leq S[i] \leq M$

SAMPLE INPUT

2
3 2
1 2
5 3
1 2 3

SAMPLE OUTPUT

Kate
Little Deepu

Note: You can reuse the elements from the set.

The Code:

```
#include <stdio.h>

int* copy;
```

```
void merge(int left, int right, int arr[]){
   int l, r, mid, iter, copyIter, diff;

   l = left;
   mid = (left + right)/ 2;
   r = mid+1;
   copyIter = 0;

   while(l<=mid && r<=right){
      if(arr[l] <= arr[r]){
         copy[copyIter++] = arr[l++];
      }
      else{
         copy[copyIter++] = arr[r++];
      }
   }

   while(l<=mid){
      copy[copyIter++] = arr[l++];
   }

   while(r<=right){
      copy[copyIter++] = arr[r++];
   }

   copyIter = 0;
   for(iter=left;iter<=right;iter++){
      arr[iter] = copy[copyIter++];
   }
}

void mergeSort(int left, int right, int arr[]){
   int mid;
   if(left < right){
      mid = (left + right) / 2;
      mergeSort(left, mid, arr);
      mergeSort(mid+1, right, arr);
      merge(left, right, arr);
   }
}



int main(){
        int caseCount;
        scanf("%d", &caseCount);

        // For item count: i indicates if Kate: 0 or Little Deepu: 1 is the winner.
        int itemWinner[10001];
        copy = (int*)calloc(100, sizeof(int));
        int set[100];
        int M, setCount, i, j, winner, diff, sorted=1;
```

```
    while(caseCount > 0) {

      scanf("%d %d", &M, &setCount);

      // take input set. Check if it's sorted.
      sorted=1;
      for(i=0; i<setCount; i++){
        scanf("%d", &set[i]);
        if(i > 0 && set[i] < set[i-1]){
          sorted=0;
        }
      }

      // If input set is not sorted, sort it.
      if(!sorted){
        mergeSort(0, setCount-1, set);
      }

      // Main logic.
      for(i=1; i<=M; i++){
        winner = 0;
        for(j=0; j<setCount; j++){
          if(i == set[j]){
            winner = 1;
            break;
          }
          else if(i > set[j]){
            diff = i - set[j];
            if(!itemWinner[diff]){
              winner = 1;
              break;
            }
          }
          else{
            break;
          }
        }
        itemWinner[i] = winner;
      }

      // print the winner.
      if(itemWinner[M]){
        printf("Little Deepu\n");
      } else {
        printf("Kate\n");
      }

      caseCount--;
    }

}
```

The Stats:

Score
30.0

Time (sec)
2.27329

Memory (KiB)
316

Language
C