13/Jun/2024

Lab-7

```
#include <stdio.h>
#include <stdlib.h>

int flag = 0;

void swap (int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}

int search (int arr[], int num, int mobile) {
    for (int g = 0; g < num; g++) {
        if (arr[g] == mobile) {
            return g + 1;
        }
    }
    return -1;
}

int find_mobile (int arr[], int d[],
    int num) {
    int mobile = 0;
    int mobile_p = 0;
    for (int i = 0; i < num; i++) {
        if (d[arr[i] - 1] == 0) && i != 0) {
            if (arr[i] > arr[i-1] && arr[i] >
                mobile_p) {
                mobile = arr[i];
                mobile_p = mobile;
```

```
        }
    }
    else if ((d[arr[i]-1]==1) &&
            i != num-1) {
        if (arr[i] > arr[i+1] &&
            arr[i] > mobile_p) {
            mobile = arr[i];
            mobile_p = mobile;
        }
    }
}
if ((mobile_p==0) && (mobile==0))
    return 0;
else
    return mobile;
}
void Permutations (int arr[], int d[],
                   int num) {
    int mobile = Find_Mobile(arr, d, num);
    int pos = Search(arr, num, mobile);
    if (d[arr[pos-1]-1]==0)
        Swap(&arr[pos-1], &arr[pos-2]);
    else
        Swap(&arr[pos-1], &arr[pos]);
    for (int i=0; i<num; i++) {
        if (arr[i] > mobile) {
            if (d[arr[i]-1]==0)
                d[arr[i]-1]=1;
```

leetcode

```
#include <stdio.h>
#include <string.h>

int cmp(const void *a, const void *b) {
    const char *pa = *(const char **)a;
    const char *pb = *(const char **)b;
    int len_a = strlen(pa);
    int len_b = strlen(pb);
    if (len_a != len_b) {
        return len_a - len_b;
    }
    return strcmp(pa, pb);
}

char *kthLongestNumber(char **nums, int numSize, int k) {
    qsort(nums, numSize, sizeof(char *), cmp);
    return nums[numSize - k];
}

int main() {
    char *nums[] = {"3", "6", "123", "23", "8"};
```

```
int numSize = 5;
int k = 2;
char* result = kthLongestNumber(nums,
                numSize, k);
printf("The %d-th longest number is: %s\n",
        k, result);
return 0;
}
```

Output:

case 1:

nums = { 3, 6, 7, 10 }

k = 4

output = "3"

nums = [2, 21, 12, 1]

k = 3

output = "2"

nums = "[0, 0]

k = 2

output = "0"

13/6/24