22/4/2024

Hackerrank code:

```
#include <iostream>
#include <algorithm>
#include <vector>

Using namespace std;

Struct Vertex {
    int left, right;
};

typedef Vector <Vertex> Vertices;

Void SwapNodes (vertices &vs, int k,
int root, int depth, bool &Short){
    if (depth % k == 0) {
    Swap (vs[root].left, vs[root].right);
    }
    if (vs [root].left != -1) {
    SwapNodes (vs, k, vs [root].left,
    depth+1, Short);
    }
    if (Short) {
        Short = false;
    } else {
        Count << " ";
    }
    Count << root;
    if (vs[root].right != -1) {
    SwapNodes (vs, k, vs [root].right,
        depth+1, Short);
    }
}
```

```
int main() {
size_t num_vertices;
cin >> num_vertices;
Vertices Vertices(num_vertices + 1);
for (size_t v = 0; v < num_vertices; ++v) {
  cin >> Vertices[v+1].left >> vertices[v+1].right;
}
size_t num_tests;
cin >> num_tests;
for (size_t t = 0; t < num_tests; ++)
{
int k;
cin >> k;
bool Start = true;
swapNodes(vertices, k, 1, 1, Start);
    count << "\n";
    }
return 0;
}
```

output:

input: 3

2 3

-1 -1

2

1

1

output: 3 1 2

~~2 1 3~~

Week - 10

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_EMPLOYEES 100
#define HASH_TABLE_SIZE 10

struct Employee {
    int key;
};
int hashFunction(int key);
void insertEmployee(struct Employee employees[],
        int hashTable[], struct Employee emp);
void displayHashTable(int hashTable[]);

int main() {
    struct Employee employees[MAX_EMPLOYEES];
    int hashTable[HASH_TABLE_SIZE] = {0};
    int n, m, i;

    printf("Enter the number of employees: ");
    scanf("%d", &n);
    printf("Enter number of employees: ");
    scanf("%d", &n);
    printf("Enter employee records: ");
    scanf("%d", &n);
    printf("Enter the number of employees: ");
    scanf("%d", &n);

    printf("Enter employee records:\n");
    for (i = 0; i < n; ++i) {
```

```
        Printf ("Employee %d : \n", i+1);
        Printf ("%d", employee[i].key);
        insert Employee (employees, hashTable, employee[i]);
    }
    Printf ("\n Hash Table: \n");
    displayHashTable (hashTable);
    return 0;
}
int hashFunction (int key) {
    return key % HASH_TABLE_SIZE;
}
Void insertEmployee (struct Employee employees[],
    int hashTable[], struct Employee emp) {
    int index = (emp.id+1) % HASH_TABLE_SIZE;
    hashTable[index] = emp.key;
}
void displayTable (int hashTable[]) {
    int i;
    for (i=0; i<HASH_TABLE_SIZE; ++i) {
        Printf ("%d --> ", i);
        if (hashTable[i] == 0) {
            Printf ("Empty \n");
        } else
            Printf ("%d \n", hashTable[i]);
    }
}
```

Output:

```
Enter the number of employees : 4
Enter Employee records:
Employee 1:
Enter 4-digit key : 550
Employee 2:
Enter 4-digit key : 300
Employee 3:
Enter 4-digit key : 550
Employee 4:
Enter 4-digit key : 376
```

Hash Table:

```
0 → 550
1 → 300
2 → 550
3 → Empty
4 → Empty
5 → 376
6 → Empty
7 → Empty
8 → Empty
9 → Empty
```

20/2/24