1. (25 points) Given a linked list defined as

   ```c
   struct node {
       int data;
       struct node *next;
   };
   ```

   Write a function defined as follows to print the index of integer “data” in this linked list. You need to print the index of the data in this linked list.

   ```c
   void search_linked_list(struct node* head, int data)
   ```
2. (25 points) Given a stack defined using linked list as follows:

```c
struct node {
    char s[50];
    struct node *next;
};
```

Write a function defined as follows to push string “b” into this stack. “count” is the number of elements in this stack. You will need create a temp node at first and return the new head node.

```c
struct node * push(struct node *head, char b[], int* count)
```
3. (25 points) Write the following function in queue. Use linked list ONLY. Function enq will make return a temp node, which can be added to the end of a queue. Function deq will release the head node memory, and return the new head node.

typedef struct node {
    char s[50];
    struct node* next;
} NODE;

NODE * enq(char b[])
{
}

NODE * deq(NODE *head)
{
}
4. (25 points) Write the insert function in binary search tree, use “iterative” method ONLY (no points for recursive method).

typedef struct Node {
    int key;
    struct Node *left;
    struct Node *right;
} Node;

Node* insert_iterative(Node* root, int key) {
    // create temp node

    // if root is NULL

    // if root is not NULL

    return root;
}