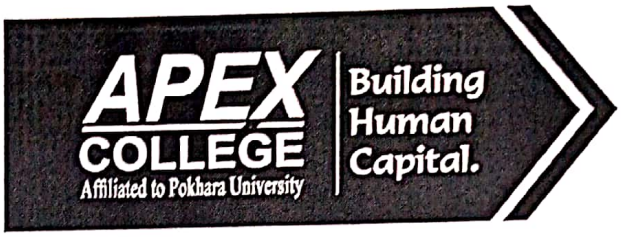
Apex College

BCIS Program

Affiliated to Pokhara University



Data Structure & Algorithms
Lab Report 9
Dynamic Implementation of Stack using Linked List

Date: 14-06-2022

Submitted by:
Ishwor Shrestha
Roll no.: 2018-BCIS-414

Submitted to:
Pravakar Ghimire, &
Anmol Shrestha
Apex College

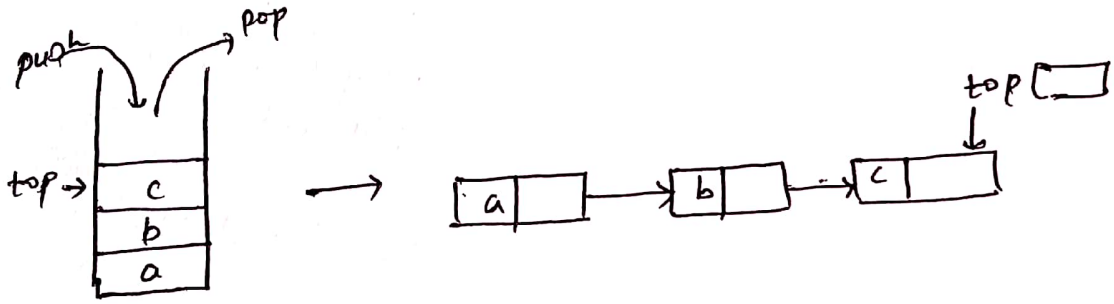
#Lab 9 Objectives

- To allocate dynamic memory performing stack operations using linked list.

#Introduction

Stack is a linear data structure that follows LIFO principles ie the item which is added at the last is removed first. In stack you can remove or add item from the top only.

We perform ~~part~~ multiple operation on stack using linked list ie. push, pop, peek, display etc.
Stack can be represent using nodes of linked list, where the very ~~first~~ last node is top from that we can add or remove nodes in a LIFO order.



#A program to implement linked list on stack data structure.

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

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

```
void push (node **top, int item){
    node *temp;
    temp = (node*) malloc (sizeof(node));
    temp->data = item;
    temp->next = *top;
    *top = temp;
}

int pop (node **top){
    int item = -1;
    node *temp;
    if (*top == NULL){
        printf("stack is an empty.\n");
    }
    else{
        temp = *top;
        *top = temp->next;
        item = temp->data;
        free(temp);
    }
    return item;
}

int peek (node **top){
    int item = -1;
    node *temp;
    if (*top == NULL)
        printf("Stack is an empty, there is no top-\n");
    else{
        temp = *top;
        *top = temp->next;
        item = temp->data;
        free(temp);
    }
    return item;
}
```

```
void display(node *top){
    node *temp;
    temp = *top;
    printf("Elements in Dynamic Stack:\n");

    while (temp != NULL) {
        printf("%d \n",temp->data);
        temp = temp -> next;
    }
}

int main() {
    node *top;
    int ch, item;
    top=NULL;
    while (1){
        printf("Enter your choice: option :");
        printf("1.PUSH \n 2.POP \n3.PEEK\n 4.DISPLAY \n5.EXIT\n");
        scanf("%d ", &ch);

        switch (ch){
            case1:
                printf("Enter data : ");
                scanf("%d", &item);
                push (&top, item);
                break;

            case2:
                item = pop (&top);
                printf("%d is popped. \n", item);
                break;
```

```
        case 3:
            item = peek(&top);
            printf("%d is top.\n", item);
            break;
        case 4:
            display(&top);
            break;
        case 5:
            exit(0);
        default:
            printf("Invalid choice.\n");
        }
    }
}
```

#Activities

In this lab, we performed various operations of stack using linked list.

1. Push operation
2. Pop operation
3. Peek operation to get value of top pointer
4. Display operation.

#Conclusion

I learned about the dynamic memory allocation in stack data structure using linked list.