Name: KAMCHE YANN ARNAUD

Matricule : FE21A208

Department: Computer Engineering

Level: 300

Task: Implement a stack using Linked List

1. CODE

```
/* Implementation of a Stack using Link List
Author: Kamche Yann Arnaud
Date: 12/05/2022
*/
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
struct Node{
	int data;
	struct Node* link;
};
struct Node* top = NULL;
//Empty Stack

void stackEmpty()
{
	if (top == NULL)
		printf("Stack: Empty");
	return;
}
//Push operations inserts a node in the stack from the beginning
void Push(int x){
	struct Node* temp = (struct Node*)malloc(sizeof(struct Node*));
	temp->data = x;
```

```
      temp->link = top;
      top = temp;
}
//Pop operaion deletes the first node
 void Pop(){
      struct Node *temp;
      if(top == NULL){
            stackEmpty();
            return;
       }
       else{

      temp = top;
      top = top->link;
      free(temp);
       return;
       }
 }
 // displays the stack to the user
void Display(){
      struct Node* temp;
      temp = top;
      printf("top-> ");
      while (temp != NULL){
            printf("%d ->", temp->data);
            temp = temp->link;
       }
       printf("NULL");
       return;
 }
 //sizeOfStack returns the number of nodes in the stack
 int sizeOfStack(){
```

```
	int count = 0;
	struct Node* temp;
	temp = top;
	while(temp != NULL){
		count++;
		temp = temp->link;
	 }
	 return count;
}
//topOfStack returns the value at the top of the stack
int topOfStack(){
	if(top == NULL)
		return -1;
	else
		return top->data;
}
int main(){
	system("color 2");
	int choice, num, Top;
	options:
	printf("\n1. Push\n");
	printf("2. Pop\n");
	printf("3. Display Element at the Top of your list\n");
	printf("4. Display your list\n");
	printf("5. Size of stack\n");
	printf("\nChoose the operation to be performed on your list: ");
	scanf("%d", &choice);

	switch(choice){
		case 1:
			printf("Enter a value: ");
			scanf("%d", &num);
```

```
                Push(num);
                break;

        case 2:
                Pop();
                break;

        case 3:
                Top = topOfStack();
                if (Top == -1)
                        printf("Top: NULL");
                else
                        printf("Top: %d", Top);
                break;
        case 4:
                Display();
                break;
        case 5:
                printf("Nodes: %d", sizeOfStack);
                break;
        default:
                break;
    }
    goto options;
    return 0;
}
```

2. COMPILATION RESULTS

I) PUSH OPERATION

```
1. Push
2. Pop
3. Display Element at the Top of your list
4. Display your list
5. Size of stack

Choose the operation to be performed on your list: 1
Enter a value: 23

Choose the operation to be performed on your list: 1
Enter a value: 45

Choose the operation to be performed on your list: 1
Enter a value: 78

Choose the operation to be performed on your list: 1
Enter a value: 56

Choose the operation to be performed on your list: 4
top-> 56 ->78 ->45 ->23 ->NULL
Choose the operation to be performed on your list: 1
Enter a value: 234

Choose the operation to be performed on your list: 4
top-> 234 ->56 ->78 ->45 ->23 ->NULL
Choose the operation to be performed on your list:
```

II) **POP OPERATION**

```
1. Push
2. Pop
3. Display Element at the Top of your list
4. Display your list
5. Size of stack

Choose the operation to be performed on your list: 2
Stack: Empty
Choose the operation to be performed on your list: 1
Enter a value: 23

Choose the operation to be performed on your list: 1
Enter a value: 45

Choose the operation to be performed on your list: 1
Enter a value: 67

Choose the operation to be performed on your list: 4
top-> 67 ->45 ->23 ->NULL
Choose the operation to be performed on your list: 2

Choose the operation to be performed on your list: 2

Choose the operation to be performed on your list: 4
top-> 23 ->NULL
Choose the operation to be performed on your list: 2

Choose the operation to be performed on your list: 4
top-> NULL
Choose the operation to be performed on your list:
```

III) Size of stack

```
1. Push
2. Pop
3. Display Element at the Top of your list
4. Display your list
5. Size of stack

Choose the operation to be performed on your list: 5
Nodes: 0
Choose the operation to be performed on your list: 1
Enter a value: 34

Choose the operation to be performed on your list: 5
Nodes: 1
Choose the operation to be performed on your list: 2

Choose the operation to be performed on your list: 5
Nodes: 0
Choose the operation to be performed on your list: 1
Enter a value: 34

Choose the operation to be performed on your list: 1
Enter a value: 56

Choose the operation to be performed on your list: 5
Nodes: 2
Choose the operation to be performed on your list: 4
top-> 56 ->34 ->NULL
Choose the operation to be performed on your list:
```

IV) Top of stack

```
1. Push
2. Pop
3. Display Element at the Top of your list
4. Display your list
5. Size of stack

Choose the operation to be performed on your list: 3
Top: NULL
Choose the operation to be performed on your list: 1
Enter a value: 23

Choose the operation to be performed on your list: 3
Top: 23
Choose the operation to be performed on your list: 1
Enter a value: 23

Choose the operation to be performed on your list: 1
Enter a value: 45

Choose the operation to be performed on your list: 1
Enter a value: 78

Choose the operation to be performed on your list: 3
Top: 78
Choose the operation to be performed on your list: 2

Choose the operation to be performed on your list: 3
Top: 45
Choose the operation to be performed on your list: 2

Choose the operation to be performed on your list: 3
Top: 23
Choose the operation to be performed on your list:
```

V) **Display stack**

```
1. Push
2. Pop
3. Display Element at the Top of your list
4. Display your list
5. Size of stack

Choose the operation to be performed on your list: 4
top-> NULL
Choose the operation to be performed on your list: 1
Enter a value: 23

Choose the operation to be performed on your list: 4
top-> 23 ->NULL
Choose the operation to be performed on your list: 1
Enter a value: 45

Choose the operation to be performed on your list: 1
Enter a value: 67

Choose the operation to be performed on your list: 1
Enter a value: 6543

Choose the operation to be performed on your list: 4
top-> 6543 ->67 ->45 ->23 ->NULL
Choose the operation to be performed on your list: 2

Choose the operation to be performed on your list: 4
top-> 67 ->45 ->23 ->NULL
Choose the operation to be performed on your list:
```