

18CSC201J  
DS ASSIGNMENT

---

DATE: 31.12.2021

---

CSE Q1 (AI-ML)

AADARSH JOSHI - RA2011026010061  
SARVESH AHUJA - RA2011026010095  
ANINDYA SHANKAR DASGUPTA - RA2011026010120

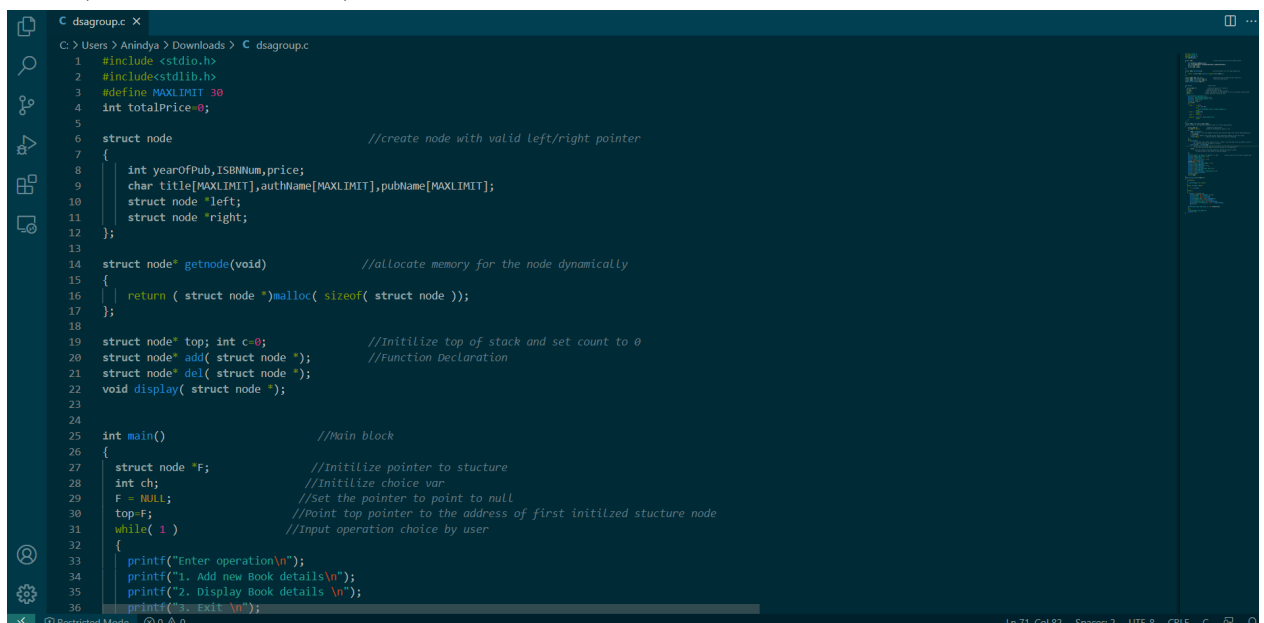
## Problem:

### Q:

1. (Q-2) Create a **stack** with 5 books using a **doubly linked list** in which the data relevant to the books such as book title, authors name, publisher name, year of publication, ISBN number, and price are stored in the nodes by creating structures.
  - i) Print the book details in Last In First Out (LIFO) order.
  - ii) Calculate the total price of the books purchased by a customer.

### A:

- Approach:
  1. It is not stated explicitly whether the data input to the stack will be user-input or predefined by the programmer. In this case, we have kept the option for user-input to facilitate more flexibility.
  2. It is not stated whether the code will be designed to handle exactly 5 inputs or more. So, we have written our code to be more scalable with little alteration of code.
  3. It is not stated whether the book details must be printed altogether, or can be printed separately. We have designed the code to accommodate flexibility regarding this.
- Code (with documentation):



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define MAXLIMIT 30
4 int totalPrice=0;
5
6 struct node
7 {
8     int yearOfPub,ISBNNum,price;
9     char title[MAXLIMIT],authName[MAXLIMIT],pubName[MAXLIMIT];
10    struct node *left;
11    struct node *right;
12};
13
14 struct node* getnode(void) //allocate memory for the node dynamically
15 {
16     return ( struct node *)malloc( sizeof( struct node ));
17 };
18
19 struct node* top; int c=0; //Initialize top of stack and set count to 0
20 struct node* add( struct node *); //Function Declaration
21 struct node* del( struct node *);
22 void display( struct node *);
23
24
25 int main() //Main block
26 {
27     struct node *F; //Initialize pointer to structure
28     int ch; //Initialize choice var
29     F = NULL; //Set the pointer to point to null
30     top=F; //Point top pointer to the address of first initialized structure node
31     while( 1 ) //Input operation choice by user
32     {
33         printf("Enter operation\n");
34         printf("1. Add new Book details\n");
35         printf("2. Display Book details \n");
36         printf("3. Exit \n");
```

```
C: dsagroup.c X
C:\> Users > Anindya > Downloads > C: dsagroup.c
34 printf("1. Add new Book details\n");
35 printf("2. Display Book details\n");
36 printf("\n. Exit\n");
37 scanf("%d", &ch);
38 switch(ch)
39 {
40     case 1: if (c<5)
41             F = add (F);
42             else
43             printf("Maximum limit of stack reached.");
44             break;
45     case 2: display(F);
46             break;
47     case 3: return 0;
48
49     default: printf("\n Wrong Choice:n");
50             break;
51 }
52 }
53
54 struct node* add( struct node *head)
55 //Return type is node Add-Func(Arguement type is of start node pointer)
56 {
57     struct node *T; //Temporary node pointer
58     if ( head == NULL ) //check if list/stack is empty or not
59     {
60         head = getNode();
61         //If its empty , call the getNode function and allocate memory for the new node dynamically
62         T =top+head;
63         //Store First address into temp pointer which represents address of the first node
64         T->left=NULL; //Back of top of Linked List points to Nothing
65     }
66     else
67     {
68         T=getNode();
69         //If already other nodes present in DLL, create a new node and store the address value in
        //the temporary pointer variable of struct*/

```

```
C: dsagroup.c X
C:\> Users > Anindya > Downloads > C: dsagroup.c
71 //The right pointer of the node at address of the top will now point
72 //to the newly allocated memory which was stored in the temporary*/
73 top=T;
74 //top now contains the Temp pointer's address(new node) so that
75 //it may be used in the future to add new nodes*/
76
77 c++;
78 printf("\nEnter the Book %d's Details: \n",c); //Store value into the newly created node
79 printf("\nEnter the title: \n");
80 scanf("%s",T->title);
81 printf("\nEnter the price: \n");
82 scanf("%ld",&T->price);
83 totalPrice += T->price;
84 printf("\nEnter the ISBN Number: \n");
85 scanf("%d",&T->ISBNNum);
86 printf("\nEnter the author:\n");
87 scanf("%s",T->authName);
88 printf("\nEnter the publisher name:\n");
89 scanf("%s",T->pubName);
90 printf("\nEnter the year of publication:\n");
91 scanf("%d",&T->yearOfPub);
92 T->right= NULL;
93 return head;
94 }
95 void display( struct node *T)
96 {
97     if(T==NULL)
98     {
99         printf("Empty\n"); return;
100     }
101     while( T->right!= NULL )
102     {
103         T = T->right;
104     }
105     if(c==5)
106     {

```

```
105 if(c==5)
106 {
107     for(int i = 5;i>0;i--){
108         printf("\nBook %d's Details: \n",i);
109         printf("Title: %s\n",T->title);
110         printf("Price: %d\n",T->price);
111         printf("ISBN Number: %d\n",T->ISBNNum);
112         printf("Author: %s\n",T->authName);
113         printf("Publisher Name: %s\n",T->pubName);
114         printf("Year of Publication: \n%d ",T->yearOfPub);
115         T=T->left;
116     }
117     printf("The total book price is: %d",totalPrice);
118 }
119 else
120 printf("Enter more Books");
121 printf("\n");
122 }
```

- Dry-run:

<u>initialization</u>	<p>First is initialization where we declare and initialize all variables and functions to be used:</p> <p>Struct node: each node of doubly-linked list of stack (contains data variables and 2 pointers - *left and *right - as required in dll.</p> <p>Then we define: top: pointer node of dll stack</p> <p>And we define functions : add and display</p>
<u>main</u>	<p>In main body,</p> <p>We give choice to user to add/display contents of dll stack</p> <p>However, it will display error if data of at least and utmost 5 books are not provided into the stack.</p> <p>Thus, we use switch case: 1 : for adding 2 : for displaying</p> <p>As shown below...</p>
<u>add</u>	<p>If switch case ch == 1</p> <p>User has to add data into dll stack</p> <p>Thus, now list is empty</p> <p>head == NULL</p> <p>T is temporary/new node getting inserted</p> <p>Thus, head points to T</p> <p>For first node, (book-1)</p> <p style="padding-left: 40px;">T-&gt;title = "Tintin"</p> <p style="padding-left: 40px;">T-&gt;price = 250</p> <p style="padding-left: 40px;">T-&gt;ISBNNum = 1</p> <p style="padding-left: 40px;">T-&gt;authName = "Hege"</p> <p style="padding-left: 40px;">T-&gt;pubName = "Penguin"</p> <p style="padding-left: 40px;">T-&gt;yearofPub = 2006</p> <p>Then T-&gt;right = NULL ; head = T (head points to first node)</p> <p style="padding-left: 40px;">T-&gt;left = NULL : first node left pointer points to NULL</p> <p style="padding-left: 40px;">top = T : top of stack, used to insert next node in order</p> <p>Similarly, all consecutive 4 nodes are inserted containing the books' details</p> <p>For each, T-&gt;right = NULL, until next node is inserted the new node points to null</p> <p>T = T-&gt;right : for inserting new node</p> <p>Thus, data for exactly 5 books are inserted, else error message is displayed.</p> <p>Last node, (book-5)</p> <p style="padding-left: 40px;">T-&gt;title = "Stars"</p> <p style="padding-left: 40px;">T-&gt;price = 250</p> <p style="padding-left: 40px;">T-&gt;ISBNNum = 5</p> <p style="padding-left: 40px;">T-&gt;authName = "Neil"</p> <p style="padding-left: 40px;">T-&gt;pubName = "BBC"</p>

	<p>T-&gt;yearofPub = 2010</p> <p>Thus, our stack is complete. Any more input leads to error message as shown in sample input/output</p> <p>While adding nodes, it simultaneously adds the price of the books and stores the sum in a variable: totalPrice</p>
<u>display</u>	<p>First, it checks if counter variable c == 5 , i.e. if all 5 data are provided, if not it displays error message.</p> <p>Then, we increment traverse pointer node T to end of list/ top of stack</p> <p>Thus, we run loop to print all book data starting from top of stack, i.e last input, thus following LIFO order as shown.</p> <p>At last, the total price is printed.</p> <p>The user may then exit from I/O console terminal.</p>

- Input/Output:

### Input-

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Anindya> cd "c:\Users\Anindya\Downloads\" ; if ($?) { gcc dsagroup.c -o dsagroup } ; if ($?) { .\dsagroup }
Enter operation
1. Add new Book details
2. Display Book details
3. Exit
1
Enter the Book 1's Details:
Enter the title:
Tintin
Enter the price:
200
Enter the ISBN Number:
1
Enter the author:
Hege
Enter the publisher name:
Penguin
Enter the year of publication:
2006
Enter operation
1. Add new Book details
2. Display Book details
3. Exit
1
Enter the Book 2's Details:
Enter the title:
Frankenstein
Enter the price:
150
Enter the ISBN Number:
2
Enter the author:
John
Enter the publisher name:
Puffin
Enter the year of publication:
2002
Enter operation

```

```
John      Enter the publisher name:
Puffin    Enter the year of publication:
2002
Enter operation
1. Add new Book details
2. Display Book details
3. Exit
1
Enter the Book 3's Details:
Enter the title:
Chronicles
Enter the price:
250
Enter the ISBN Number:
3
Enter the author:
Gary
Enter the publisher name:
Penguin
Enter the year of publication:
2010
Enter operation
1. Add new Book details
2. Display Book details
3. Exit
1
Enter the Book 4's Details:
Enter the title:
Inception
Enter the price:
300
Enter the ISBN Number:
4
Enter the author:
Henry
Enter the publisher name:
Penguin
Enter the year of publication:
2015
Enter operation
1. Add new Book details
```

```
Enter the publisher name:
Penguin
Enter the year of publication:
2015
Enter operation
1. Add new Book details
2. Display Book details
3. Exit
1
Enter the Book 5's Details:
Enter the title:
Stars
Enter the price:
250
Enter the ISBN Number:
5
Enter the author:
Neil
Enter the publisher name:
BBC
Enter the year of publication:
2010
Enter operation
1. Add new Book details
2. Display Book details
3. Exit
1
Maximum limit of stack reached.Enter operation
1. Add new Book details
2. Display Book details
3. Exit
2
```

## Output-

```
Book 5's Details:
Title: Stars
Price: 250
ISBN Number: 5
Author: Neil
Publisher Name: BBC
Year of Publication:
2010
Book 4's Details:
Title: Inception
Price: 300
ISBN Number: 4
Author: Henry
Publisher Name: Penguin
Year of Publication:
2015
Book 3's Details:
Title: Chronicles
Price: 250
ISBN Number: 3
Author: Gary
Publisher Name: Penguin
Year of Publication:
2010
Book 2's Details:
Title: Frankenstein
Price: 150
ISBN Number: 2
Author: John
Publisher Name: Puffin
Year of Publication:
2002
Book 1's Details:
Title: Jintin
Price: 200
ISBN Number: 1
Author: Hege
Publisher Name: Penguin
Year of Publication:
2006
The total book price is: 1150
```

- Result:

1. Time Complexity:

Other elements take  $O(1)$  to execute.

In the main block:

while(1) has time complexity =  $O(1)$

If we consider there are  $n$  nodes in total then,

In display function:

while loop to reach the last node :  $O(n)$

In the for condition, to print  $n$  nodes :  $O(n)$

Total Time Complexity:  $n+n+1+1 = 2n+2$   
 $= O(n)$

2. Space Complexity:

For the struct node: 3 int data type variables =  $4*3 = 12$  bytes

3 char data type variables =  $1*3 = 3$  bytes

Node \*left and \*right = 2 bytes

Similarly for other constant elements we have const. memory allocated =  $x$  bytes

If we have  $n$  nodes in the DLL, then space occupied by the list is  $= n*(12+3+2) = 17n$  bytes

Total Space complexity :  $(17n + x)$  bytes =  $O(n)$

- Division of work:

- Aadarsh Joshi (RA2011026010061) -
    - Code logic, Code documentation, Time/space complexity of program
  - Sarvesh Ahuja (RA2011026010095)-
    - Code logic, Writing code in VScode, Editing and debugging code
  - Anindya Shankar Dasgupta (RA2011026010120)-
    - Code logic, Sample test/run of code in compiler for input-output, Approach, Dry-run.
-