[DATA STRUCTURES IN C - MCA B - 2022](https://classroom.google.com/c/NDc1NTM4OTgyMzU5" \t "_self)

LabProgram\_01

Question 01:  Implement Stack using Array

//Array implementation of stack with the example of adding books on a shelf

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#define SIZE 10

void push(int);

void pop();

void display();

int stack[SIZE], top = -1;

int main()

{

int value, choice;

while(1){

printf("\n\n\*\*\*BOOK SHELF\*\*\*\n");

printf("1. Add Books\n2.Remove Books \n3.Show books\n4. Exit");

printf("\nChoose one of the following: ");

scanf("%d",&choice);

switch(choice){

case 1: printf("Enter the number of books to be added: ");

scanf("%d",&value);

push(value);

break;

case 2: pop();

break;

case 3: display();

break;

case 4: exit(0);

default: printf("\nTry again!");

}

}

}

void push(int value){ //Function to insert the number of

if(top == SIZE-1)

printf("\nBook shelf is Full! Can't insert anymore'");

else{

top++;

stack[top] = value;

printf("\nBooks are added!");

}

}

void pop(){

if(top == -1)

printf("\nBook Shelf is Empty!");

else{

printf("\nDeleted : %d", stack[top]);

top--;

}

}

void display(){

if(top == -1)

printf("\nBook Shelf is Empty!!!");

else{

int i;

printf("\nThe number of books on the shelf are:\n");

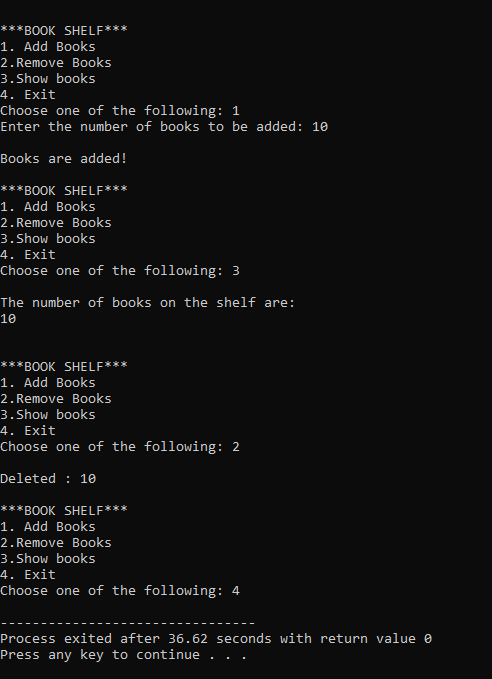
for(i=top; i>=0; i--)

printf("%d\n",stack[i]);

}

}

Output:



Question 02:  Implement Stack using Linked List

#include <stdio.h>

#include <stdlib.h>

struct node

{

int info;

struct node \*ptr;

}

\*tos,\*tos1,\*temp;

void push(int data);

void pop();

void empty();

void display();

int peek();

int count = 0;

int main()

{

int num, ch, ele;

printf("\n 1 - Push");

printf("\n 2 - Pop");

printf("\n 3 - Top");

printf("\n 4 - Empty");

printf("\n 5 - Exit");

printf("\n 6 - Display");

while (1)

{

printf("\n Enter choice : ");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("Enter data : ");

scanf("%d", &num);

push(num);

break;

case 2:

pop();

break;

case 3:

if (tos == NULL)

printf("No elements in stack");

else

{

ele = peek();

printf("\n Top element : %d", ele);

}

break;

case 4:

empty();

break;

case 5:

exit(0);

case 6:

display();

break;

default :

printf(" Wrong choice, Please enter correct choice ");

break;

}

}

}

/\* Push data into stack \*/

void push(int data)

{

if (tos == NULL)

{

tos =(struct node \*)malloc(1\*sizeof(struct node));

tos->ptr = NULL;

tos->info = data;

}

else

{

temp =(struct node \*)malloc(1\*sizeof(struct node));

temp->ptr = tos;

temp->info = data;

tos = temp;

}

count++;

}

void display()

{

tos1 = tos;

if (tos1 == NULL)

{

printf("Stack is empty");

return;

}

while (tos1 != NULL)

{

printf("%d ", tos1->info);

tos1 = tos1->ptr;

}

}

void pop()

{

tos1 = tos;

if (tos1 == NULL)

{

printf("\n Error : Trying to pop from empty stack");

return;

}

else

tos1 = tos1->ptr;

printf("\n Popped value : %d", tos->info);

free(tos);

tos = tos1;

count--;

}

int peek()

{

return(tos->info);

}

void empty()

{

if (tos == NULL)

printf("\n Stack is empty");

else

printf("\n Stack is not empty with %d elements", count);

}

Output:

