\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Data Structure Lab

CEN-391

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Program 6(b)

Code :-

#include <iostream>

using namespace std;

struct stack

{

    int data;

    stack \*next;

} \* top;

void Display()

{

    cout << "Display...\n";

    if (top == nullptr)

    {

        cout << "Stack Is Empty" << endl;

        return;

    }

    cout << "\n";

    stack \*temp = top;

    while (temp != nullptr)

    {

        cout << temp->data << " ";

        temp = temp->next;

    }

    cout << "\n";

}

void Push()

{

    cout << "Push...\n";

    stack \*newnode = (stack \*)malloc(sizeof(stack));

    if (newnode == nullptr)

    {

        cout << "Stack Overflow" << endl;

        return;

    }

    cout << "Enter The Number : ";

    cin >> newnode->data;

    newnode->next = top;

    top = newnode;

    cout << "\n";

    Display();

}

void Pop()

{

    cout << "Pop...\n";

    if (top == nullptr)

    {

        cout << "Stack Underflow" << endl;

        return;

    }

    cout << top->data << "\n";

    stack \*todelete = top;

    top = top->next;

    delete todelete;

    cout << "\n";

    Display();

}

void Top()

{

    cout << "Top...\n";

    if (top == nullptr)

    {

        cout << "Stack Is Empty" << endl;

        return;

    }

    cout << top->data << "\n";

}

void isEmpty()

{

    cout << "isEmpty...\n";

    if (top != nullptr)

    {

        cout << "Not Empty \n";

    }

    else

    {

        cout << "Empty \n";

    }

}

void Total\_Elements()

{

    cout << "Total Elements...\n";

    int total = 0;

    stack \*temp = top;

    while (temp != nullptr)

    {

        total++;

        temp = temp->next;

    }

    cout << total << "\n";

}

void Bars()

{

    cout << "---------------------------------------------------------------\n";

}

int Options()

{

    int opt;

    cin >> opt;

    Bars();

    switch (opt)

    {

    case 1:

        Push();

        break;

    case 2:

        Pop();

        break;

    case 3:

        isEmpty();

        break;

    case 4:

        Top();

        break;

    case 5:

        Total\_Elements();

        break;

    case 6:

        Display();

        break;

    case 7:

        cout << "Exit...\n";

        return 0;

    default:

        cout << "Invalid Input!\nTry Again!\n";

    }

    Bars();

    return 1;

}

void Menu()

{

    cout << "\_\_\_\_\_Operations\_On\_Stacks\_\_\_\_\_ \n";

    cout << "1.Push \n";

    cout << "2.Pop \n";

    cout << "3.isEmpty \n";

    cout << "4.Top \n";

    cout << "5.Total Elements \n";

    cout << "6:Display \n";

    cout << "7.Exit \n";

    cout << "Enter Your Choice : ";

}

int main()

{

    system("cls");

    cout << "\_\_\_\_\_Vicky\_Gupta\_20BCS070\_\_\_\_\_\n\n";

    while (true)

    {

        Menu();

        if (!Options())

            break;

    }

    cout << "Exiting...\n";

    Bars();

    return 0;

}

Output :-

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated