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

Data Structure Lab

CEN-391

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

Program 6(a)

Code :-

#include <iostream>

using namespace std;

int size;

struct stack

{

    int \*arr;

    int top;

} st;

void Display()

{

    cout << "Display...\n";

    if (st.top == -1)

    {

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

        return;

    }

    cout << "\n";

    for (int i = 0; i <= st.top; i++)

    {

        cout << st.arr[i] << " ";

    }

    cout << "\n";

}

void Push()

{

    cout << "Push...\n";

    if (st.top == size - 1)

    {

        cout << "Stack Overflow" << endl;

        return;

    }

    st.top++;

    int val;

    cout << "Enter The Number : ";

    cin >> val;

    st.arr[st.top] = val;

    cout << "\n";

    Display();

}

void Pop()

{

    cout << "Pop...\n";

    if (st.top == -1)

    {

        cout << "Stack Underflow" << endl;

        return;

    }

    cout << st.arr[st.top] << "\n";

    st.top--;

    cout << "\n";

    Display();

}

void Top()

{

    cout << "Top...\n";

    if (st.top == -1)

    {

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

        return;

    }

    cout << st.arr[st.top] << "\n";

}

void isEmpty()

{

    cout << "isEmpty...\n";

    if (st.top != -1)

    {

        cout << "Not Empty \n";

    }

    else

    {

        cout << "Empty \n";

    }

}

void isFull()

{

    cout << "isFull...\n";

    if (st.top+1 == size)

    {

        cout << "Full \n";

    }

    else

    {

        cout << "Not Full \n";

    }

}

void Total\_Elements()

{

    cout << "Total Elements In Stack...\n";

    cout << st.top + 1 << "\n";

}

void Bars()

{

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

}

int Options()

{

    int opt;

    cin >> opt;

    Bars();

    switch (opt)

    {

    case 1:

        Push();

        break;

    case 2:

        Pop();

        break;

    case 3:

        isFull();

        break;

    case 4:

        isEmpty();

        break;

    case 5:

        Top();

        break;

    case 6:

        Total\_Elements();

        break;

    case 7:

        Display();

        break;

    case 8:

        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.isFull \n";

    cout << "4.isEmpty \n";

    cout << "5.Top \n";

    cout << "6.Total Elements \n";

    cout << "7:Display \n";

    cout << "8.Exit \n";

    cout << "Enter Your Choice : ";

}

int main()

{

    system("cls");

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

    cout << "Enter The Size Of The Stack : ";

    cin >> size;

    st.arr = (int \*)malloc(size \* sizeof(int));

    st.top = -1;

    cout << "\n\n";

    while (true)

    {

        Menu();

        if (!Options())

            break;

    }

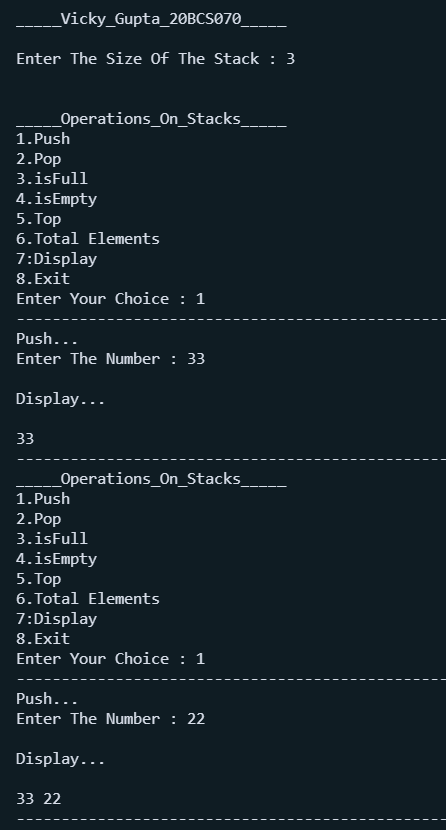
    cout << "Exiting...\n";

    Bars();

    return 0;

}

Output :-



Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated