Ex 4: Stack Implementation

PROGRAM :

#include <stdio.h>

#include <stdlib.h>

struct Node

{

int Data;

struct Node \*next;

}\*top;

void popStack()

{

struct Node \*temp, \*var=top;

if(var==top)

{

top = top->next;

free(var);

}

else

printf("\nStack Empty");

}

void push(int value)

{

struct Node \*temp;

temp=(struct Node \*)malloc(sizeof(struct Node));

temp->Data=value;

if (top == NULL)

{

top=temp;

top->next=NULL;

}

else

{

temp->next=top;

top=temp;

}

}

void display()

{

struct Node \*var=top;

if(var!=NULL)

{

printf("\nElements are as:\n");

while(var!=NULL)

{

printf("\t%d\n",var->Data);

var=var->next;

}

printf("\n");

}

else

printf("\nStack is Empty");

}

int main()

{

int i=0;

top=NULL;

clrscr();

printf(" \n1. Push to stack");

printf(" \n2. Pop from Stack");

printf(" \n3. Display data of Stack");

printf(" \n4. Exit\n");

while(1)

{

printf(" \nChoose Option: ");

scanf("%d",&i);

switch(i)

{

case 1:

{

int value;

printf("\nEnter a value to push into Stack: ");

scanf("%d",&value);

push(value);

break;

}

case 2:

{

popStack();

printf("\n The last element is popped");

break;

}

case 3:

{

display();

break;

}

case 4:

{

struct Node \*temp;

while(top!=NULL)

{

temp = top->next;

free(top);

top=temp;

}

exit(0);

}

default:

{

printf("\nwrong choice for operation");

}}}}

OUTPUT:

