AIM:

Write a c program to implement a stack using array and linked list implementation and execute the following operation on stack.

(i)Push an element into a stack

(ii) Pop an element from a stack

(iii) Return the Top most element from a stack

(iv)  Display the elements in a stack

ALGORITHM:

I. start

2. Create a structure and functions for the given operations.

3. Initialize Stack array with capacity and top=1.

4. To push an element into a stack read Itu data to be pushed. If the top is equal to capacity-1 display stack over flow otherwise increment the top and push the data onto Stack at index top.

5. To pop an element from a Stack if the is equal to-I display as stack underflow. Alter wise pop delta from Stack at index fop the decrement the top and display the popped data.

6. To return the pop most element from a Stack it the top is equal to-1 display Stack is empty. Otherwise display all elements in stack from top too.

7. After these operations display all elements in Stack from top too.

8. End.

PROGRAM:

#include <stdio.h>

#define size 100

int stack[size];

int top = -1, i;

void push() {

int a;

printf("Enter the data to insert in stack : ");

scanf("%d", &a);

if (top == size - 1) {

printf("Stack is full\n");

} else {

top = top + 1;

stack[top] = a;

}

}

void pop() {

int a;

if (top == -1) {

printf("The stack is empty\n");

} else {

a = stack[top];

printf("Item popped is : %d\n", a);

top--;

}

}

void peek() {

if (top == -1) {

printf("The stack is empty\n");

} else {

printf("The top most element in the stack is : %d\n", stack[top]);

}

}

void display() {

if (top == -1) {

printf("Stack is empty\n");

} else {

printf("Items in the stack are : ");

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

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

printf("\n");

}

}

int main() {

int choice;

printf("STACK IMPLEMENTATION USING ARRAY\n");

printf("1.Push\n2.Pop\n3.Peek\n4.Display\n5.Exit\n");

do {

printf("Enter Your Choice : ");

scanf("%d", &choice);

switch (choice) {

case 1:

push();

break;

case 2:

pop();

break;

case 3:

peek();

break;

case 4:

display();

break;

case 5:

printf("Exiting!!\n");

break;

default:

printf("Invalid choice\n");

}

} while (choice != 5);

return 0;

}

#include <stdio.h>

#include <stdlib.h>

struct Node {

int data;

struct Node\* next;

};

struct Node\* top = NULL;

void push(int x) {

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

if (temp == NULL) {

printf("Stack Overflow\n");

return;

}

temp->data = x;

temp->next = top;

top = temp;

}

void pop() {

if (top == NULL) {

printf("Stack Underflow\n");

return;

}

struct Node\* temp = top;

top = top->next;

free(temp);

}

int peek() {

if (top == NULL) {

printf("Stack is empty\n");

return -1;

}

return top->data;

}

void display() {

struct Node\* temp = top;

printf("Stack: ");

while (temp != NULL) {

printf("%d ", temp->data);

temp = temp->next;

}

printf("\n");

}

int main() {

int choice, item;

printf("STACK IMPLEMENTATION USING LINKED LIST\n");

printf("1. Push\n2. Pop\n3. Peek\n4. Display\n5. Exit\n");

do {

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter element to push: ");

scanf("%d", &item);

push(item);

break;

case 2:

pop();

break;

case 3:

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

break;

case 4:

display();

break;

case 5:

printf("Exiting...\n");

break;

default:

printf("Invalid choice\n");

}

} while (choice != 5);

return 0;

}

OUTPUT:

Stack implemptation using array.

1. push

2. pop

3. Peak

4. Display

2. Exit.

Enter your choice:1

Enter your data to input: 12

Enter your choice:

Enter the data to input: 23

Enter your choice :3

Ite hop most element: 23

Enter your Choice: 4

Stack: 231L

Enter your choice: 2

Item popped is 23

Enter your Choice : 4

Stack:12

Enter your Choice:5

Exiting!

RESULT:

This code is verified and implemented successfully.