1. MERGE STACK

AIM:

To merge two stacks to the stack 1.

ALGORITHM:

1. Start
2. Read user input for both stacks
3. In the first stack, push in through user input, while keeping track of the last element using a variable (say top).
4. With the second stack, push to the stack 1, beginning from the top+1 position from the last stack operated on.
5. Print the merged stack (stack 1).
6. End

DIAGRAMMATIC REPRESENTATION:

Stack 1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 75 | 24 |  |  |  |  |

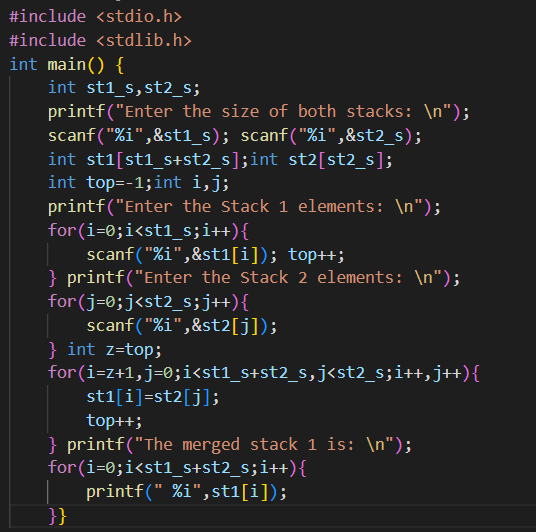
Stack 2:

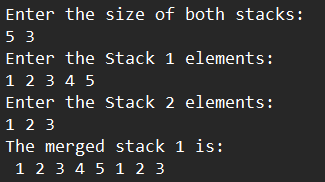
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 41 | 99 |  |  |  |

Stack 1 (final):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 75 | 24 | 2 | 41 | 99 |  |

SAMPLE INPUT/OUTPUT:





RESULT:

The merging of two stack elements to a single stack is successfully implemented using pop() and push() operations on the stack.

1. STACK SPLITTING

AIM:

To split Stack S into stacks S1 and S2 such that S1 contains only positive numbers and S2 contains only negative numbers

ALGORITHM:

1. Start
2. Receive input of elements for stack S, along with a variable for keeping track of the last element in the stack (say top\_org).
3. Initialize two stacks for storing positive and negative numbers; declare two variables (say top\_1 and top\_2) for tracking last elements respectively.
4. Traverse through the original stack, pop the negative values and positive values and push them to positive number stack and negative number stack respectively.
5. Update the top\_1 and top\_2 variables respectively.
6. Print the positive and negative stack respectively using the top variables.
7. End

DIAGRAMMATIC REPRESENTATION:

Original Stack:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | -6 | 2 | -7 | 3 | -8 | 4 | -9 | 5 | 10 |

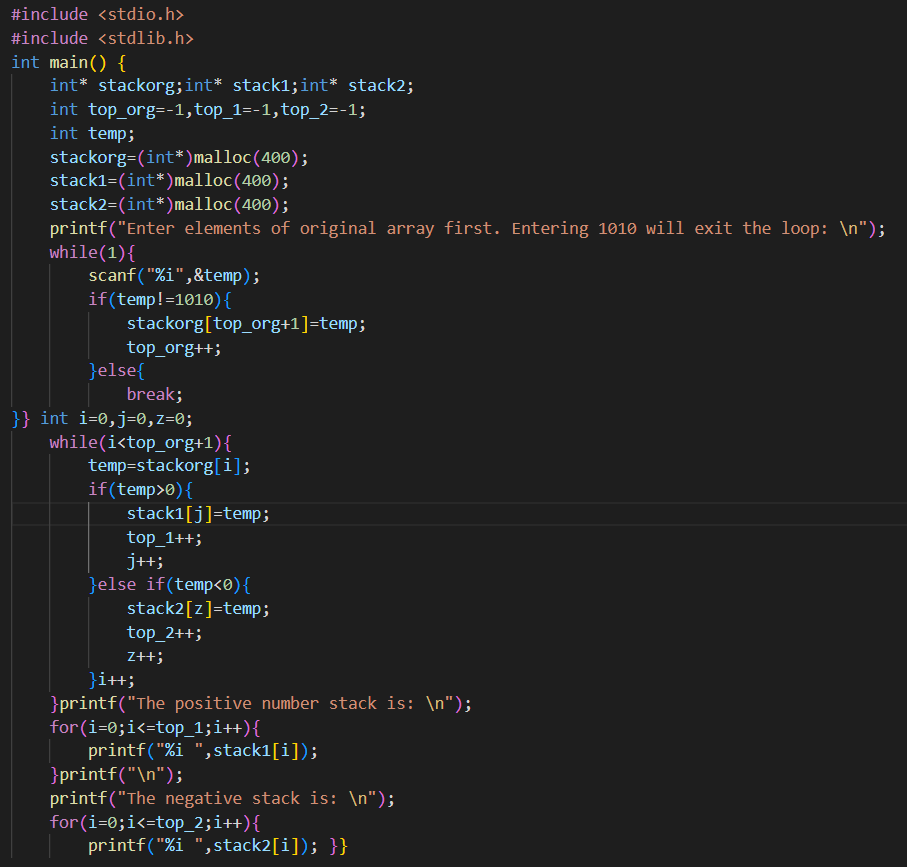
Positive number stack:

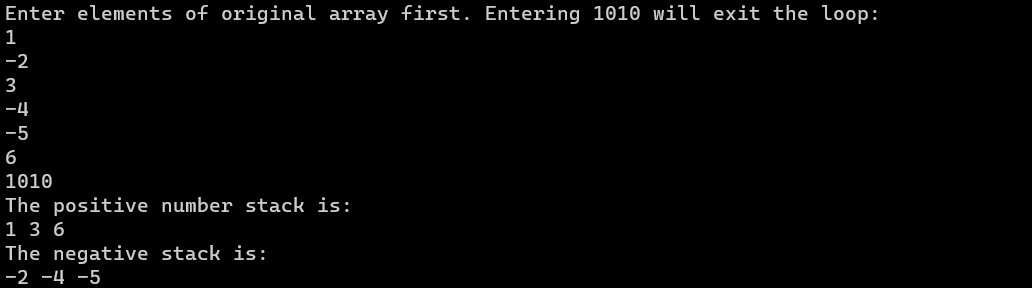
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |  |  |  |  |  |

Negative number stack:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| -6 | -7 | -8 | -9 | -10 |  |  |  |  |  |

SAMPLE INPUT/OUTPUT:





Result:

The stack splitting to positive and negative numbers is implemented successfully using the stack operations push() and pop().

|  |  |  |  |
| --- | --- | --- | --- |
| Observation | Code and Output | Viva | Record |
|  |  |  |  |