The Stack

A data structure in which items are added and removed only from one end (the "top")
A program must set aside a block of memory to hold the stack by declaring a stack segment

```
stack 256
```

SS will contain the segment number of the stack segment -- SP will be initialized to 256 (100h)
The stack grows from higher memory addresses to lower ones

PUSH and POP

New words are added with push
push *source*
SP is decreased by 2
a copy of the source contents is moved to SS:SP
Items are removed with pop
pop *destination*
Content of SS:SP is moved to the destination
SP is increased by 2

Stack example

```
push ax ;Save ax and bx
push bx ; on the stack
mov ax, -1 ;Assign test values
mov bx, -2
mov cx, 0
mov dx, 0
push ax ;Push ax onto stack
push bx ;Push bx onto stack
pop cx ;Pop cx from stack
pop dx ;Pop dx from stack
pop bx ;Restore saved ax and bx
pop ax ; values from stack
```