

```
#include <stdio.h>
#include <stdlib.h>

/* Beispiel fuer Stack, der int-Werte in einen Vektorspeichern kann */
static int *stack = NULL, /* Stack als int-Vektor */
          sp=0, /* Stack Pointer (akt. Index) zeigt auf freies Element */
          unten=0, /* Index untere Grenze */
          oben=0; /* Index obere Grenze */

/* Anlegen Stack mit n>0 Elementen */
void stackinit(int n) {
    if(n>0){
        stack = (int*)malloc(n*sizeof(int)); /* dynamisches Anlegen auf dem heap */
        sp = 0;
        unten = 0;
        oben = n-1;
    }
}

int isEmpty() { return sp <= unten; } /* Stack leer ? */

int isFull() { return sp > oben; } /* Stack voll ? */

void push(int wert) { /* wert im Stack speichern */
    if (stack == NULL) { /* ex. Stack nicht ? */
        printf("Stack fehlt\n");
        return;
    }
    if (isFull()) { /* Stack ist voll ? */
        printf("Stack voll\n");
        return;
    }
    stack[sp++] = wert; /* stack[sp]=wert, sp=sp+1 */
}

void pop() { /* Entferne oberstes Element aus Stack */
    if (stack == NULL) {
        printf("Stack fehlt\n");
        return;
    }
    if (isEmpty()) {
        printf("Stack leer\n");
        return;
    }
    --sp; /* Stackpointer um 1 reduzieren */
}

/* Rueckgabe, Lesen oberstes Element, ohne zu loeschen */
int top() {
    if (stack == NULL) {
        printf("Stack fehlt\n");
        return -1;
    }
}
```

```

    if (isEmpty()) {
        printf("Stack leer\n");
        return -1;
    }
    return stack[sp-1];
}

int popTop() { /* oberstes Element lesen und entfernen */
    int wert = top();
    pop();
    return wert;
}

int main(void) {
    int z=0, n=0;

    printf("Stackgroesse: ");
    scanf("%d", &n); /* Groesse n des Stacks einlesen */
    while(getchar() != '\n');

    stackinit(n); /* Stack mit n Elementen anlegen */

    while (1) { /* unendliche Schleife */
        printf("int-Zahl fuer Stack (F6=EOF): ");
        scanf("%d", &z); /* Wert fuer Stack einlesen */
        while(getchar() != '\n');

        if(feof(stdin)) /* Abbruch bei F6 bzw. Strg+Z (EOF) */
            break;
        push(z); /* Zahl z auf Stack legen */
        z=0;
    }
    printf("Inhalt Stack der Groesse %d :\n", n);

    while (!isEmpty()) /* Solange Stack nicht leer */
        printf("%d, ", popTop()); /* oberstes Element entfernen */
    printf("\n");
    getchar();
    return 0;
}
/*
Stackgroesse: 4
int-Zahl fuer Stack (F6=EOF): 0
int-Zahl fuer Stack (F6=EOF): 1
int-Zahl fuer Stack (F6=EOF): 2
int-Zahl fuer Stack (F6=EOF): 3
int-Zahl fuer Stack (F6=EOF): 4
Stack voll
int-Zahl fuer Stack (F6=EOF): ^Z

Inhalt Stack der Groesse 4 :
3, 2, 1, 0,
*/

```