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

typedef struct node{
    void* data;
    struct node* next;
}Node;

typedef struct stack{
    Node* head;
    unsigned int size;
}Stack;

Stack* init();

Node* initNode();

void push(Stack* stack, void* data);

void pop(Stack* s, void (*delete) (void*
data));

void* peek(Stack* s);

void destroy(Stack* s, void (*delete)
(void* data));

int size(Stack* s);
```

```c
#include "stack.h"

Stack* init(){
    Stack* s = malloc(sizeof(Stack));
    s-> head = NULL;
    s-> size = 0;
    return s;
}

Node* initNode(){
    Node* n = malloc(sizeof(Node));
    if(n){
        n-> data = NULL;
        n-> next = NULL;
    }
    return n;
}

void push(Stack* stack, void* data){
    if(stack){
        Node* node = initNode();
        if(node){
            node-> data = data;
            node-> next = stack-> head;
            stack-> head = node;
            stack-> size++;
        }
    }
}

void pop(Stack* stack, void (*delete)
(void* data)){
    if(stack){
        Node* tmp = stack-> head;
        if(tmp){
            stack-> head = tmp-> next;
            stack-> size --;
            delete(tmp-> data);
            free(tmp);
            tmp = NULL;
        }
    }
}
```

```c
void* peek(Stack* stack){
    if(stack){
        return stack-> head-> data;
    }
    return NULL;
}

void destroy(Stack* stack, void (*delete)
(void* data)){
    if(stack){
        Node* current;
        while(current = stack-> head){
            delete(current-> data);
            stack-> head = current-> next;
            free(current);
        }
        free(stack);
        stack = NULL;
    }
}

int size(Stack* s){
    if(s){
        return s-> size;
    }
    return -1;
}
```