Implementing the Stack ADT Using a Fixed-Length Array

There are several ways to implement the stack ADT. Our first approach is the simplest. We'll have the `stackADT.c` file define the `stack_type` structure so that it contains a fixed-length array (to hold the contents of the stack) along with an integer that keeps track of the top of the stack:

```
struct stack_type {
  int contents[STACK_SIZE];
  int top;
};
```

Here's what `stackADT.c` will look like:

stackADT.c

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

#define STACK_SIZE 100

struct stack_type {
  int contents[STACK_SIZE];
  int top;
};

static void terminate(const char *message)
{
  printf("%s\n", message);
  exit(EXIT_FAILURE);
}

Stack create(void)
{
  Stack s = malloc(sizeof(struct stack_type));
  if (s == NULL)
    terminate("Error in create: stack could not be created.");
  s->top = 0;
  return s;
}

void destroy(Stack s)
{
  free(s);
}

void make_empty(Stack s)
{
  s->top = 0;
}

bool is_empty(Stack s)
{
  return s->top == 0;
}
```