9 Stack with characters

Modify the stack example so that it stores characters instead of integers. Next, add a `main` function that asks the user to enter a series of parentheses and/or braces, then indicates whether or not they're properly nested:

```
Enter parentheses and/or braces: (){}{()}
Parentheses/braces are nested properly
```

Hint: As the program reads characters, have it `push` each left parenthesis or left brace. When it reads a right parenthesis or brace, have it `pop` the stack and check that the item popped is a matching parenthesis or brace. (If not, the parentheses/braces aren't nested properly.) When the program reads the new-line character, have it check whether the stack is empty; if so, the parentheses/braces are matched. If the stack **isn't** empty (or if `stack_underflow` is ever called), the parentheses/braces aren't matched. If `stack_overflow` is called, have the program print the message `Stack overflow` and terminate immediately.

Solution

```
/*********************************************************/
// stack.c
// Solution to Programming Project 1 (Chapter 10)
// From C PROGRAMMING: A MODERN APPROACH (Second Edition)
// Copyright (c) 2008 W. W. Norton & Company
// All rights reserved.
// This program may be freely distributed for class use,
// provided that this copyright notice is retained.
/*********************************************************/

#include <stdbool.h>   /* C99 only */
#include <stdio.h>
#include <stdlib.h>

#define STACK_SIZE 100

/* external variables */
char contents[STACK_SIZE];
int top = 0;
```

```
/* prototypes */
void make_empty(void);
bool is_empty(void);
bool is_full(void);
void push(char ch);
char pop(void);
void stack_overflow(void);
void stack_underflow(void);

int main(void)
{
  bool properly_nested = true;
  char ch;

  printf("Enter parentheses and/or braces: ");
  while (properly_nested && (ch = getchar()) != '\n')
    if (ch == '(' || ch == '{')
      push(ch);
    else if (ch == ')')
      properly_nested = !is_empty() && pop() == '(';
    else if (ch == '}')
      properly_nested = !is_empty() && pop() == '{';

  if (properly_nested && is_empty())
    printf("Parentheses/braces are nested properly\n");
  else
    printf("Parentheses/braces are NOT nested properly\n");

  return 0;
}

void make_empty(void)
{
  top = 0;
}

bool is_empty(void)
{
  return top == 0;
}
```

```
bool is_full(void)
{
  return top == STACK_SIZE;
}

void push(char ch)
{
  if (is_full())
    stack_overflow();
  else
    contents[top++] = ch;
}

char pop(void)
{
  if (is_empty())
    stack_underflow();
  else
    return contents[--top];

  return '\0'; /* prevents compiler warning due to stack_underflow() call */
}

void stack_overflow(void)
{
  printf("Stack overflow\n");
  exit(EXIT_FAILURE);
}

void stack_underflow(void)
{
  printf("Stack underflow\n");
  exit(EXIT_FAILURE);
}
```