

The background features a white surface with scattered, colorful abstract shapes. These include several yellow triangles of varying sizes, some light blue curved lines, and some light green curved lines. On the left side, there are larger, more complex shapes in shades of purple and lavender, some resembling stylized swirls or loops. The overall aesthetic is clean and modern, typical of a presentation slide.

C Programming Basic – week 12

A decorative background on the left side of the slide features a light green balloon at the top, a light blue balloon in the middle, and a light purple balloon at the bottom. Yellow streamers and triangular confetti are scattered around the balloons.

Topics of this week

- Recursive functions
- Exercises

Exercise 12.1

- Write a recursive algorithm for dealing a deck of cards. The parameters should be (i) the deck of undealt cards, and (ii) the person who is to receive the next card.
Assuming:
 - The players seat around a table
 - Start with the player on the left of the dealer
 - Each step involves dealing one card to a player, then moving to the next player on the left
 - Dealing continues until no card is left in the deck

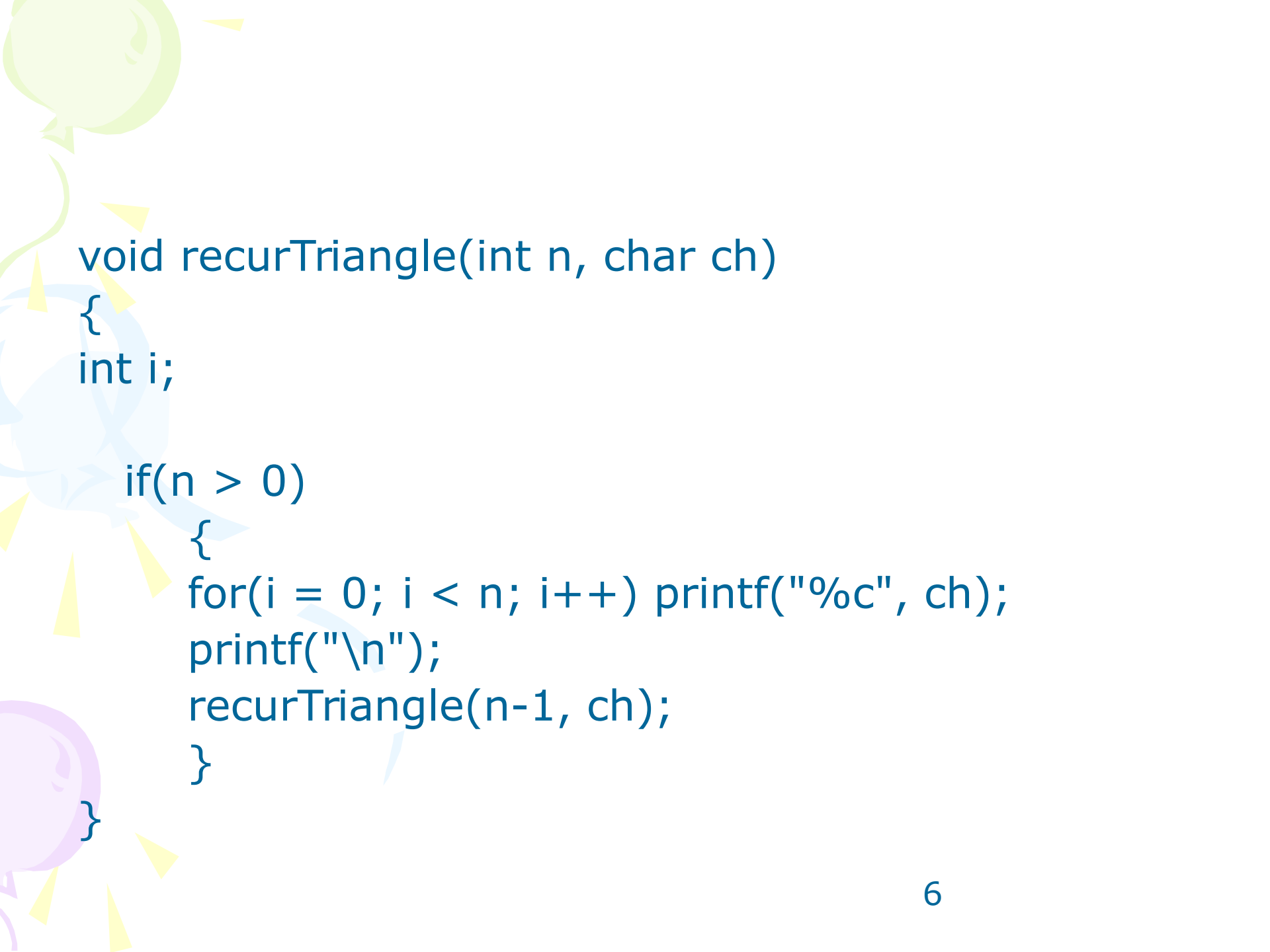
Hint

```
function dealCards (deck, person)
{
  if (deck is empty)
    return;
  deal top card from deck to person;
  dealCards (rest of deck, personLeftOf(person));
}
```

Exercise 12.2

- Write a recursive function **void recurTriangle (int n, char ch)** which prints out an upside-down triangle. The parameter *ch* is the character to be used for drawing the triangle, and *n* is the number of characters on the first row. For example, if *n* is 7 and *ch* is '+', then the output of the function should be:

```
+++++++  
+++++++  
+++++++  
+++++  
++++  
+++  
++  
+
```

The background features a light green balloon in the top left, a purple balloon in the bottom left, and several yellow streamers and triangular flags scattered across the page.

```
void recurTriangle(int n, char ch)
{
    int i;

    if(n > 0)
    {
        for(i = 0; i < n; i++) printf("%c", ch);
        printf("\n");
        recurTriangle(n-1, ch);
    }
}
```

Exercise 12.3

- Write a function “preceding()” to compare two strings in dictionary order

```
int preceding(char *first, char *second)
```

- Return value
 - 1 if *first* is “smaller than” *second*
 - 0 if *first* is equal to *second*
 - -1 if *first* is “greater than” *second*

Exercise 12.4

- Write a function `setup_nameList()` to read the names of n persons ($2 \leq n \leq 25$) from a file and put them into an array `nameList[]` of string

```
int setup_nameList(char *nameList[], char
*filename)
```


Exercise 12.5

- Write a function “qsort_name()” to sort an array of strings in dictionary order using Quicksort

Homework 1

- Write a function to sort a singly linked list using Quicksort. Add this function to the linked list library.
- Hint: You should have functions
 - *To get the n^{th} element in the list*
 - *To swap two nodes in the list*

The slide features a decorative background on the left side with a light green balloon at the top, a light blue balloon in the middle, and a light purple balloon at the bottom. Yellow streamers and triangular flags are scattered around the balloons. The main content is centered on the right side of the slide.

Improving Quicksort

- Change the Pivot Selection strategy:
 - random element
 - median of three strategy