

The background features a white surface with scattered, colorful abstract shapes. These include thin, curved lines in shades of light blue, light green, and light purple. Interspersed among these are numerous small, yellow, triangular shapes that resemble confetti or paper scraps. The overall aesthetic is clean and modern.

# **C Programming Basic – week 12**

# Nội dung

- Hàm đệ quy
- Bài tập

# Exercise 12.1

- Viết hàm đệ quy để chia bài. Các tham số bao gồm (i) các quân bài chưa được chia, and (ii) người tiếp theo được chia bài. Giả sử:
  - Người chơi ngồi xung quanh bàn
  - Bắt đầu bằng người chơi ở bên trái của người chia bài
  - Tại mỗi bước chia bài cho một người sau đó chuyển sang người tiếp theo ở bên trái
  - Chia bài đến khi không còn lại quân bài nào

# Gợi ý

```
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

- Viết hàm đệ quy **void recurTriangle (int n, char ch)** in ra hình tam giác. Tham số *ch* là kí tự dùng để phủ hình tam giác, và *n* là số kí tự ở dòng đầu tiên. VD, nếu  $n = 7$  và *ch* là '+', thì in ra hình sau:

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

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 flags are scattered around the balloons.

```
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

- Viết hàm “preceding()” để so sánh hai chuỗi theo thứ tự từ điển

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

- Giá trị trả về:
  - 1 nếu *first* > *second*
  - 0 nếu *first* = *second*
  - -1 nếu *first* < *second*

# Exercise 12.4

- Viết hàm "setup\_nameList()" đọc tên của  $n$  người ( $2 \leq n \leq 25$ ) từ tệp và lưu vào mảng các chuỗi "nameList[]"

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



# Exercise 12.5

- Viết hàm “qsort\_name()” để sắp xếp các chuỗi theo thứ tự từ điển sử dụng giải thuật sắp xếp nhanh

# Homework 1

- Viết hàm sắp xếp danh sách liên kết đơn sử dụng giải thuật sắp xếp nhanh. Thêm hàm này vào thư viện danh sách liên kết
- Gợi ý: Cần có các hàm sau
  - Lấy phần tử thứ  $n$  trong danh sách
  - Đổi chỗ hai phần tử trong danh sách

# Cải thiện sắp xếp nhanh

- Thay đổi chiến thuật lựa chọn pivot
  - Lựa chọn ngẫu nhiên
  - Phần tử trung vị của 3 phương pháp