

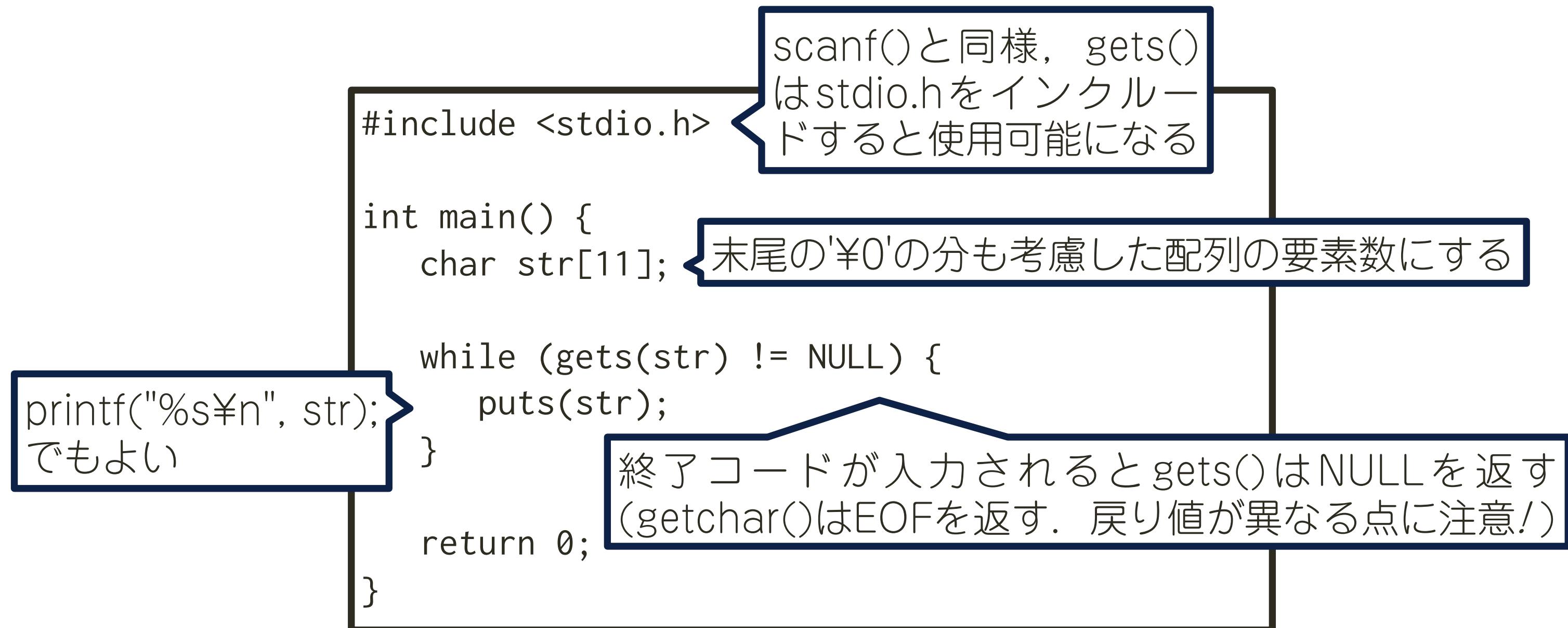
# 前回課題の解説

## (第10回: 文字列)

担当: 佐藤

2018年12月3日(月)

# 課題10-1



- scanf()の%s変換指定子で文字列を読み込む場合, 空白文字を読み込むことができない. そのため, この問題では文字列の読み込みにgets()を使う必要がある

# 課題10-2

strcat() を使うために  
string.h をインクルード

iStr に入力された文字  
列を cStr に連結し、連  
結後の cStr を表示する  
ことを繰り返す

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

```
int main() {
    char cStr[101] = "";
    char iStr[21];
```

```
    while (gets(iStr) != NULL) {
        strcat(cStr, iStr);
        puts(cStr);
    }
```

```
    return 0;
```

```
}
```

strcat() は連結先の文字配列の '\0' の位置に連結元の文字列の先頭の文字を連結する。したがって、**連結先の文字配列には '\0' がなければならない**。この条件を満たすため、通常、連結先の文字配列はあらかじめ空文字列にしておく。この処理は初期化で簡単にできる

実は、strcat() の戻り値は連結後の文字列。そのため、まとめて1行で「puts(strcat(cStr, iStr));」と書くこともできる

# 課題10-3

```
#include <stdio.h>
#include <string.h> {strcmp()を使うために必要}

int main() {
    char list[10][20] = { "Australia", "Austria", "Denmark", "Dubai", "Egypt",
                          "France", "Germany", "Hawaii", "Italy", "Russia" };

    char str[10];
    int i;

    printf("行きたい旅行先は?¥n");
    scanf("%s", str); {「gets(str);」でもよい}
    for (i = 0; i < 10; i++) {
        if (strcmp(list[i], str) == 0) {
            printf("%sはリストの%d番目です¥n", str, i + 1);
            break; {無駄にループしないよう, 番号調整}
        }
    }
    if (i == 10) {
        printf("%sはリストにありません¥n", str);
    }

    return 0;
}
```

無駄にループしないだけでなく  
i == 10を「見つからなかった場合」の条件  
として使えるようになるという効果もある

# 課題10-4(1)

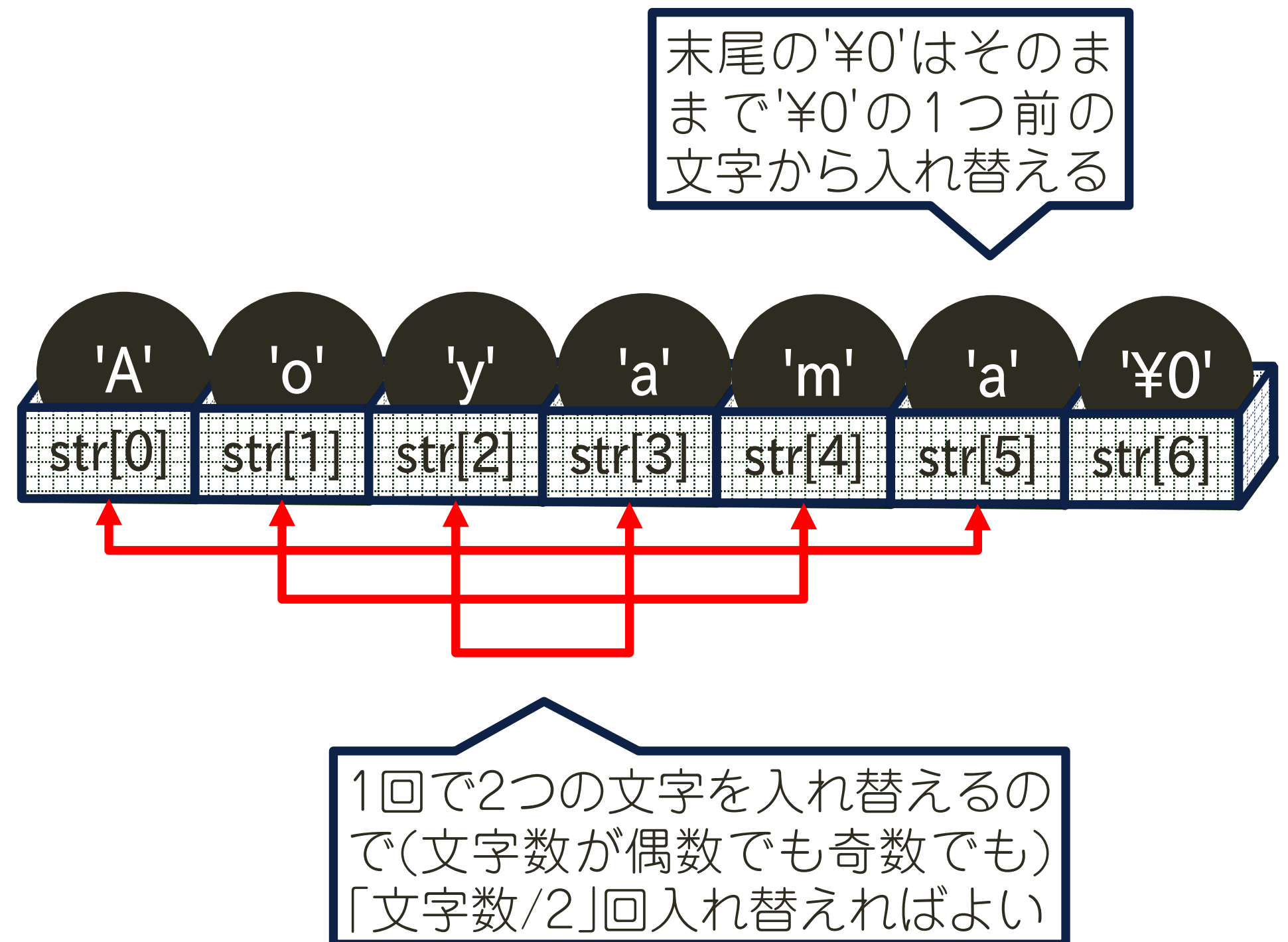
## 文字配列の要素数に依存するプログラム

```
#include <stdio.h>

int main() {
    char str[7];
    char tmp;
    int i;

    printf("文字列を入力してください\n");
    scanf("%s", str);
    for (i = 0; i < 7 / 2; i++) {
        tmp = str[i];
        str[i] = str[5 - i];
        str[5 - i] = tmp;
    }
    printf("%s\n", str);

    return 0;
}
```



# 課題10-4(2)

文字配列の要素数に依存しない汎用的なプログラム

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

int main() {
    char str[128], tmp;
    int i, j;

    puts("文字列を入力してください");
    gets(str);
    for (i = 0, j = strlen(str); i < --j; i++) {
        tmp = str[i];
        str[i] = str[j];
        str[j] = tmp;
    }
    puts(str);

    return 0;
}
```

strlen(), カンマ演算子および前置減分演算子を利用する