

# Cプログラミング 入門

— プログラミング基礎 (3) : 条件分岐 —

早稲田大学

# 今回の目標

- **条件分岐**が適切に使える。
  - if else 文
  - switch 文
- **論理演算**が理解でき，使える
  - かつ (AND) 「&&」
  - または (OR) 「||」
  - 否定 (NOT) 「!」

# 例題 1

課題：体格指数 BMI を計算するプログラムを作れ

BMI は  $BMI = 10^4 \times (\text{体重 [kg]}) \div (\text{身長 [cm]})^2$  で与えられる低体重・肥満などを判定する医学的指数である．表示は以下のようにし，ファイル名は bmi1.c とせよ．

*Input your height[cm]: 173.2【Enter】*

*Input your weight[kg]: 60.3【Enter】*

*Your BMI is 20.1.*

- $10^4 = 10000.0$
- ディレクトリ移動を忘れずに
- ファイル編集は
- コンパイルは
- 実行は

# 例題 1

課題：体格指数 BMI を計算するプログラムを作れ

BMI は  $BMI = 10^4 \times (\text{体重 [kg]}) \div (\text{身長 [cm]})^2$  で与えられる低体重・肥満などを判定する医学的指数である．表示は以下のようにし，ファイル名は bmi1.c とせよ．

*Input your height[cm]: 173.2 【Enter】*

*Input your weight[kg]: 60.3 【Enter】*

*Your BMI is 20.1.*

- $10^4 = 10000.0$
- ディレクトリ移動を忘れずに 【cd work】
- ファイル編集は
- コンパイルは
- 実行は

# 例題 1

## 課題：体格指数 BMI を計算するプログラムを作れ

BMI は  $BMI = 10^4 \times (\text{体重 [kg]}) \div (\text{身長 [cm]})^2$  で与えられる低体重・肥満などを判定する医学的指数である．表示は以下のようにし，ファイル名は bmi1.c とせよ．

*Input your height[cm]: 173.2 【Enter】*

*Input your weight[kg]: 60.3 【Enter】*

*Your BMI is 20.1.*

- $10^4 = 10000.0$
- ディレクトリ移動を忘れずに 【cd work】
- ファイル編集は 【emacs bmi.c &】
- コンパイルは
- 実行は

# 例題 1

## 課題：体格指数 BMI を計算するプログラムを作れ

BMI は  $BMI = 10^4 \times (\text{体重 [kg]}) \div (\text{身長 [cm]})^2$  で与えられる低体重・肥満などを判定する医学的指数である．表示は以下のようにし，ファイル名は bmi1.c とせよ．

*Input your height[cm]: 173.2 【Enter】*

*Input your weight[kg]: 60.3 【Enter】*

*Your BMI is 20.1.*

- $10^4 = 10000.0$
- ディレクトリ移動を忘れずに 【cd work】
- ファイル編集は 【emacs bmi.c &】
- コンパイルは 【gcc bmi.c】
- 実行は

# 例題 1

## 課題：体格指数 BMI を計算するプログラムを作れ

BMI は  $BMI = 10^4 \times (\text{体重 [kg]}) \div (\text{身長 [cm]})^2$  で与えられる低体重・肥満などを判定する医学的指数である．表示は以下のようにし，ファイル名は bmi1.c とせよ．

*Input your height[cm]: 173.2 【Enter】*

*Input your weight[kg]: 60.3 【Enter】*

*Your BMI is 20.1.*

- $10^4 = 10000.0$
- ディレクトリ移動を忘れずに 【cd work】
- ファイル編集は 【emacs bmi.c &】
- コンパイルは 【gcc bmi.c】
- 実行は 【./a.out】

## 例題1のヒント

- 変数宣言：

```
double Height, Weight, BMI;
```

- 表示：printf(" ")

- 身長入力：scanf(" ")

```
scanf("%lf",&Height);
```

- BMIの計算：

```
BMI = 10000*Weight / (Height*Height);
```

- BMIの出力：

```
printf("%f", BMI);
```



## if 文 (条件分岐)

### if 文の使い方

```
if (条件式) {  
    条件式が真のときに実行させたいこと  
}else{  
    条件式が偽のときに実行させたいこと  
}
```

- 実行させたいことが1つの場合は、それを囲む {} は省略可。
- else 以下は、やらせたいことがなければ省略可。
- 入れ子にして複雑な条件分岐を設定することも可能。
- インデント (行頭の字下げ) は美しく設定する。

# if 文

例：

```
if (x>0) printf("x is a positive number %fn");  
if (x>0) {  
    printf("x is a positive number %fn");  
}
```

# if 文

## 比較演算子【==, !=, <, >, <=, >=】

- **条件式**では、**比較演算子**を用いて条件となる式を記述する。
- 真ならば1，偽ならば0の値を取る。

$a==b$	a と b が等しければ 1
$a!=b$	a と b が等しくなければ 1
$a<b$	a が b より小さければ 1
$a>b$	a が b より大きければ 1
$a<=b$	a が b より小さいもしくは等しければ 1
$a>=b$	a が b より大きいもしくは等しければ 1

- x の値が 10 に等しい : `if (x==10) ...`  
注意 : 「=」は**代入**を表し、「==」は**等しい**を表す。
- x の値が 20 以上 : `if (x>=20) ...`
- $a*b+c < a/b$  など計算式にも使える

# if 文

## 論理演算子【||, &&, !】

- 条件式では、**論理演算子**を用いて複数の条件を満たす記述が可能。

<code>a    b</code>	a と b のうち少なくとも片方が真なら 1
<code>a &amp;&amp; b</code>	a と b のうち両方が真なら 1
<code>! a</code>	a が真なら 0, 偽なら 1

- x の値が 10 以上 20 未満：  
`if (x >= 10 && x < 20) ...`

# 論理演算

## 【真】と【偽】の二種類の値のみで行う演算

- 「 $x \geq 3 \ \&\& \ x < 5$ 」:  $x$  が 3 以上, **かつ**  $x$  が 5 未満で【真】
- 「 $x \geq 3 \ \|\| \ x < 5$ 」:  $x$  が 3 以上, **または**  $x$  が 5 未満で【真】  
注) 専門用語の『または』は両方真のときも【真】となる
- 「 $!(x == 3)$ 」: ( $x$  が 3 に等しい) でないと真,  
注) 「 $x != 3$ 」と同じ意味

## 複雑な例

- 「 $!(x == 0 \ \|\| \ y == 0)$ 」: 「 $x != 0 \ \&\& \ y != 0$ 」と同じ  
注 1) 優先順位が紛らわしい場合は ( ) をつけるとよい  
注 2) ( ) をつけないと「!」、「<」、「>」、「<=」、「>=」、「==」、「!=」、「&&」、「!!」の順に行われる

## if 文使用プログラム例 1

```
#include <stdio.h>
int main(void){
    double Score = 85;

    if(                Score<80) printf("Your grade is B¥n");
    if(80<=Score && Score<90) printf("Your grade is A¥n");
    if(90<=Score                ) printf("Your grade is A++¥n");

    return 0;
}
```

- 変数 Score の値により、右の文字列を表示：

80未満	Your grade is B
80以上90未満	Your grade is A
90以上	Your grade is A++

- 「&&」は左右両側の条件がともに真となった場合に真となる。

## if 文使用プログラム例 2

```
#include <stdio.h>
int main(void){
    double Score = 85;
    if(Score < 80)
        printf("Your grade is B\n");
    else if(Score < 90)
        printf("Your grade is A\n");
    else
        printf("Your grade is A++\n");
    return 0;
}
```

- このような書き方でも同様に表示 .
- if 文が入れ子になっていることに注意 :

80未満	Yes	Your grade is B		
	No	90未満	Yes	Your grade is A
			No	Your grade is A++

## if文使用プログラム例3

```
if(Score< 70){
    if(Score< 60) printf("Your grade is D¥n");
    else          printf("Your grade is C¥n");
}else{
    if(Score< 80) printf("Your grade is B¥n");
    else{
        if(Score< 90) printf("Your grade is A¥n");
        else          printf("Your grade is A++¥n");
    }
}
```

- 見た目は複雑だが，一度に処理ができる：

70未満	Yes	60未満	Yes	Your grade is D		
			No	Your grade is C		
	No	80未満	Yes	Your grade is B		
			No	90未満	Yes	Your grade is A
No	No	Your grade is A++				



## 例題 2

課題：BMI の WHO 基準を表示させよ。

授業のはじめに作ったプログラムを改良して，次の WHO 基準を表示させるプログラムを作れ（ファイル名は bmi2.c）

BMI	WHO 基準
18.5 未満	Underweight
18.5 以上 25.0 未満	Normal
25.0 以上 30.0 未満	Pre-obese
30.0 以上	Obese class

表示は以下のようにする。

*Input your height[cm]: 173.2 【Enter】*

*Input your weight[kg]: 60.3 【Enter】*

*Your BMI is 20.1. Normal.*

# switch 文

## switch 文の使い方

```
switch (整数式){  
    case 整数定数式 1 :  
        やらせたいこと 1 ;  
        break ;  
    case 整数定数式 2 :  
        やらせたいこと 2 ;  
        break ;  
    case 整数定数式 3 :  
        やらせたいこと 3 ;  
        break ;  
    default :  
        やらせたいこと 4 ;  
        break ;  
}
```

- switch 文の後の ( ) 内の**整数式の値**によって、やらせたいことを場合分けすることができる (実数はダメ)
- 一致する case 文 (なければ default 文) が実行される
- default 文は省略可
- 「break;」がないと default 文を続けて実行してしまう。「**break;**」を入れること

## switch 文使用プログラム例

```
#include <stdio.h>
int main(void){
    int num;
    scanf("%d", &num);
    printf("%d is divisible by 3 with",num);

    switch (num % 3){
        case 1 : printf("remainder 1¥n"); break;
        case 2 : printf("remainder 2¥n"); break;
        default: printf("remainder 0¥n"); break;
    }

    return 0;
}
```

- 「num % 3」は3で割ったときの余りを求める。
- 入力した値を3で割った余りによって、異なる文字列を表示。

## 演算子の優先順位（参考）

1	式のグループ化「()」, 関数呼出「()」, 配列添字「[]」, メンバー選択「->」,「.」
2	論理否定「!」, ビット反転「~」, 符号「+」「-」, 前置・後置「++」「--」, 参照「&」, 間接参照*, サイズ取得「sizeof」
3	乗算「*」, 除算「/」, 剰余「%」
4	加算「+」, 減算「-」
5	ビットシフト「<<」,「>>」
6	大小関係「<」,「>」,「<=」,「>=」
7	等号「==」, 不等号「!=」
8	ビット AND「&」
9	ビット XOR「^」
10	ビット OR「 」
11	論理 AND「&&」
12	論理 OR「  」
13	条件選択「?:」
14	単純代入「=」, 混合代入「*=」,「/=」,「%=」,「+=」, 「-=」,「&=」,「^=」,「 =」
15	カンマ区切り「,」

# まとめ

- 条件分岐

- if else 文
- switch 文

- 論理演算

- かつ (AND) 「&&」
- または (OR) 「||」
- 否定 (NOT) 「!」