

初心者のための C言語講座

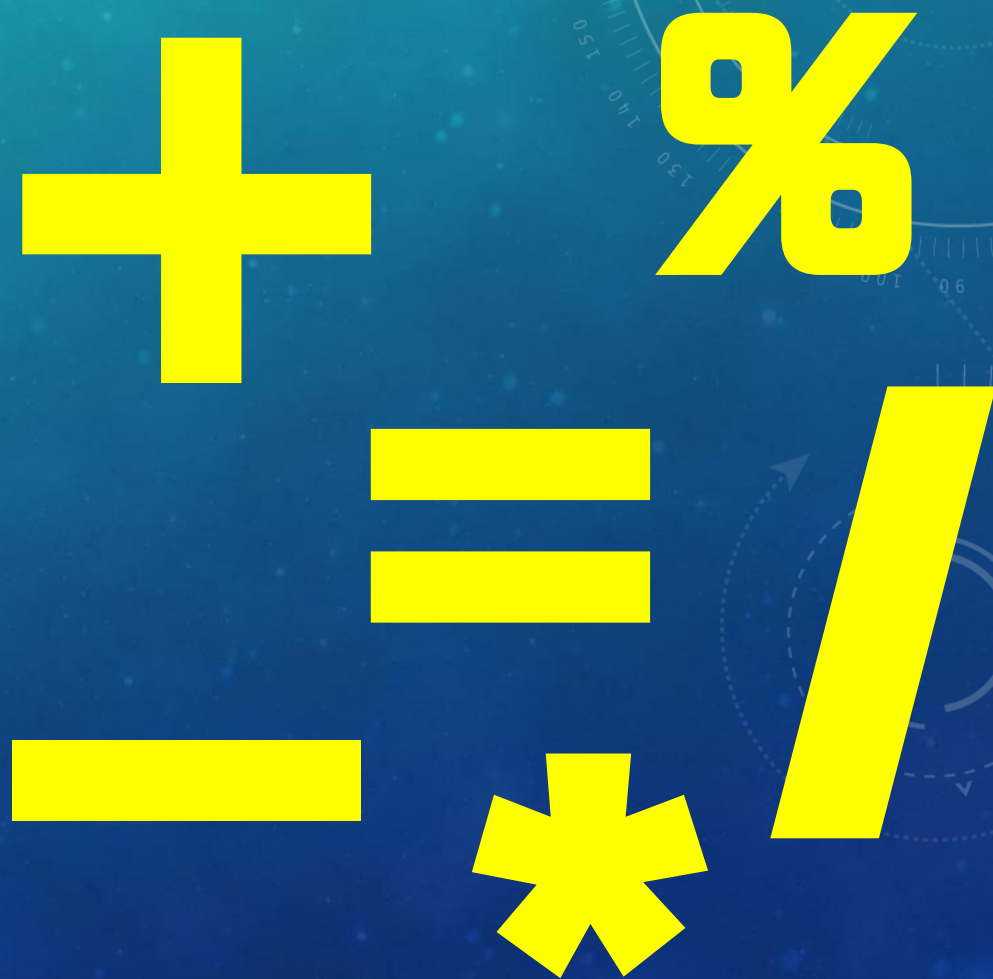
#4: if文, switch文

Column) マツコの知らない演算子の世界

性懲りもなくコラムから始まる講座。

次のページ以降に、C言語で主に扱うことになる演算子をリストアップしました。なぜ、急にこんなコラムを最初に持ってきたのかというと、**演算子はif文やswitch文で多用するから**です。

既に教えてあるものもありますが、一通りチェックしておいてくださいね。



Column) マツコの知らない演算子の世界

① 算術演算子

演算子 (赤字部分)	意味	
$a + b$	加算	$a + b$
$a - b$	減算	$a - b$
$a * b$	乗算	$a \times b$
a / b	除算	$a \div b$
$a \% b$	剰余	aをbで割ったときの余り

▲ Figure4.1 主な算術演算子

Column) マツコの知らない演算子の世界

② 代入演算子

演算子 (赤字部分)	意味	
$a = b$	代入	aにbを代入する
$a += b$	加算代入	「 $a = a + b$ 」と同じ意味
$a -= b$	減算代入	「 $a = a - b$ 」と同じ意味

▲ Figure4.2 主な代入演算子

Column) マツコの知らない演算子の世界

③ 比較演算子

演算子 (赤字部分)	意味	
$a == b$	等価	a と b は等しい
$a != b$	非等価	a と b は等しくない
$a > b$	超過	$a > b$
$a < b$	未満	$a < b$
$a >= b$	以上	$a \geq b$
$a <= b$	以下	$a \leq b$

▲ Figure4.3 主な比較演算子

Column) マツコの知らない演算子の世界

④ 論理演算子

演算子 (赤字部分)	意味	
a && b	論理AND	aかつb (aもbも真ならば真)
a b	論理OR	aまたはb (a, bのどちらか一方でも真ならば真)

▲ Figure4.4 主な論理演算子

Column) マツコの知らない演算子の世界

⑤ その他の演算子

演算子 (赤字部分)	意味	
<code>a++</code>	インクリメント	「 <code>a = a + 1</code> 」と同じ意味
<code>a--</code>	デクリメント	「 <code>a = a - 1</code> 」と同じ意味

▲ Figure4.5 その他の演算子

Column) マツコの知らない演算子の世界

初心者の方に、特に注意してほしい点があります。

「=」と「==」の区別はしっかりしてください。プログラムにおいて「=」は代入、「==」は等価の意味です。

数学に引っ張られて、「 $x = 5$ 」と見たときに、「 x は5と等しいのか！」なんて言わないこと。あくまで「=」は代入なので、正しくは「 x に5を代入する」という意味になります。

さあ、次のページからが本題ですよ！

【数学】
 x は5と等しい

$$x = 5$$

【プログラム】
 x に5を代入する

#4-1) if文

演算子はバッチリですか？ 忘れてしまったら、適宜確認しましょう。では、本題のif文に入っていきます。

if文は、以下のように使用します。

if(条件式) 指示;

これは条件式を満たしたときにだけ、その指示を行うという意味です。右のProgram4.1を例に挙げると、条件式が「 $x \geq 10$ 」となっているので、入力された数字が10以上ならば「10以上の数です。」と表示します。10未満ならば何もせずにプログラムを終了するようになっています。

```
#include <stdio.h>

int main(void){
    int x;
    scanf("%d", &x);
    if(x >= 10) printf("10以上の数です。¥n");
    return 0;
}
```

▲ Program4.1 ten_over.c

```
26 [Enter]
10以上の数です。
```

▲ Result4.1 ten_over.cの実行結果例

#4-1) if文

先ほどは、指示が1つだけだったので1行で済みましたが、指示がたくさんある場合はどうしましょうか？

そういうときは、以下のように指示全体を{}で括ればOKです。

```
if(条件式) {
```

```
    指示;
```

```
    指示;
```

```
    . . . . .
```

```
}
```

```
#include <stdio.h>

int main(void){
    int x;
    scanf("%d", &x);
    if(x >= 10){
        printf("10以上の数です。¥n");
        printf("すっごーい!¥n");
    }
    return 0;
}
```

▲ Program4.2 ten_over2.c

```
26 [Enter]
10以上の数です。
すっごーい!
```

▲ Result4.2 ten_over2.cの実行結果例

#4-1) if文

さて、いきなりですが問題です。右の空欄には何を入れればいいのでしょうか？

「xが10以上20以下」という条件式を作ればいいので……。

```
#include <stdio.h>

int main(void){
    int x;
    scanf("%d", &x);
    if( )
        printf("10以上20以下の数です。¥n");
    return 0;
}
```

▲ Program4.3 ten_twenty.c

#4-1) if文

「`10 <= x <= 20`」と答えた方、いらっしゃいますね。

おめでとうございます。違います。

C言語では、このような表現は認められていません。したがって、「`10 <= x && x <= 20`」というように、分割して&&でまとめる必要があります。勘違いする人が多いので気を付けてくださいね。

```
#include <stdio.h>

int main(void){
    int x;
    scanf("%d", &x);
    if(10 <= x && x <= 20)
        printf("10以上20以下の数です。¥n");
    return 0;
}
```

▲ Program4.3 ten_twenty.c

#4-2) if-else

if文だけだと、条件式を満たさなかった場合の処理はされません。条件式を満たさない場合もフォローしているのが、if-else文です。使い方は以下の通り。

if(条件式) 条件式を満たすときの指示;

else 条件式を満たさないときの指示;

if文のあとに、else～を追加で書くだけです。簡単！
ちなみに、if-else文についても{}で括れば複数の指示を出すことが可能です。

```
#include <stdio.h>

int main(void){
    int x;
    scanf("%d", &x);
    if(x >= 10){
        printf("10以上の数です。¥n");
        printf("すっごーい!¥n");
    }
    else{
        printf("10未満の数です。¥n");
        printf("たーのしー!¥n");
    }
    return 0;
}
```

▲ Program4.4 ten_over3.c

```
5 [Enter]
10未満の数です。
たーのしー!
```

▲ Result4.4 ten_over3.cの実行結果例

#4-2) if-else

if-else文を用いれば、次のような芸当もできます。

右のProgram4.5は入力された点数に対して、成績を表示するプログラムです。「A+」「A」「B」「C」「F」の5段階の評価をつけなければならないので、素直に考えるとこのようにif文を5個使うことになります。

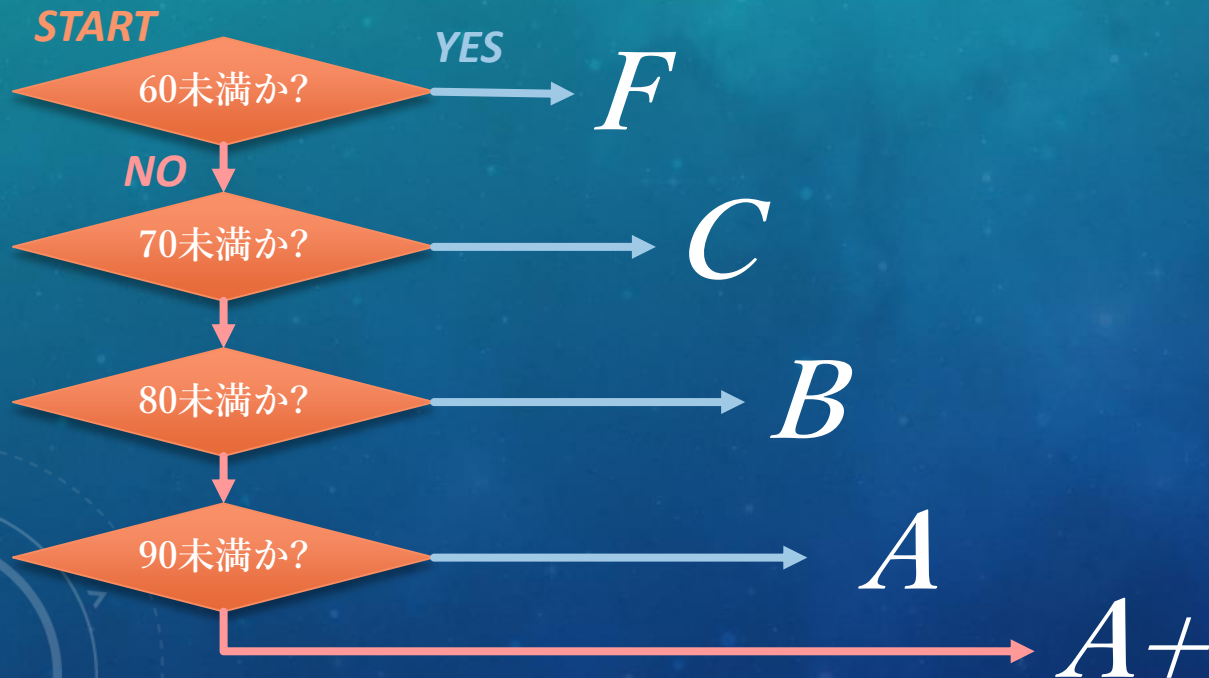
```
#include <stdio.h>

int main(void){
    int point;
    scanf("%d", &point);
    if(point < 60) printf("F¥n");
    if(60 <= point && point < 70) printf("C¥n");
    if(70 <= point && point < 80) printf("B¥n");
    if(80 <= point && point < 90) printf("A¥n");
    if(90 <= point) printf("A+¥n");
    return 0;
}
```

▲ Program4.5 grade.c

#4-2) if-else

このプログラムをif-else文で書き直したものが、右に示したProgram4.6です。条件文がスッキリして見やすくなりました。このように、elseの後にifを何個も接続していくことができるのです。



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

```
int main(void){  
    int point;  
    scanf("%d", &point);  
    if(point < 60) printf("F¥n");  
    else if(point < 70) printf("C¥n");  
    else if(point < 80) printf("B¥n");  
    else if(point < 90) printf("A¥n");  
    else printf("A+¥n");  
    return 0;  
}
```

▲ Program4.6 grade2.c

#4-3) switch文

さて、switch文の話に移りましょう。switch文は右のProgram4.7のような感じで使われます。

次のページにswitch文の使い方を、一般化したものを載せておいたので、頑張って理解してみてください。

```
#include <stdio.h>

int main(void){
    int x;
    scanf("%d", &x);
    switch(x % 3){
        case 0:
            printf("xを3で割ると余りは0¥n");
            break;
        case 1:
            printf("xを3で割ると余りは1¥n");
            break;
        default:
            printf("xを3で割ると余りは2¥n");
    }
    return 0;
}
```


#4-3) switch文

```
switch(式){  
    case 定数1:  
        式の結果が定数1のときに行う処理;  
        . . . . .  
        break;  
    case 定数2:  
        式の結果が定数2のときに行う処理;  
        . . . . .  
        break;  
    default:  
        式の結果がどの定数でもないときに行う処理;  
        . . . . .  
}
```

```
#include <stdio.h>  
  
int main(void){  
    int x;  
    scanf("%d", &x);  
    switch(x % 3){  
        case 0:  
            printf("xを3で割ると余りは0¥n");  
            break;  
        case 1:  
            printf("xを3で割ると余りは1¥n");  
            break;  
        default:  
            printf("xを3で割ると余りは2¥n");  
    }  
    return 0;  
}
```

#4-3) switch文

何点か注意事項を挙げます。

①

caseは何個あっても構いません。

②

caseの最後には、**break**を忘れないこと。

③

defaultは必須というわけではないのですが、可能な限り書いておいた方が良いでしょう。(予期せぬ入力があったときにバグるので)

```
#include <stdio.h>

int main(void){
    int x;
    scanf("%d", &x);
    switch(x % 3){
        case 0:
            printf("xを3で割ると余りは0¥n");
            break;
        case 1:
            printf("xを3で割ると余りは1¥n");
            break;
        default:
            printf("xを3で割ると余りは2¥n");
    }
    return 0;
}
```

Column) コメントアウト

すごく今更感がありますが、コメントアウトについて触れておきます。

コメントというのは、それ自体はプログラムに一切干渉しない文字列のことです。「そんなの何に使うんだ」と考える人もいるかと思いますが、実際は結構な頻度で使います。

コメントでプログラムの説明を書いたり、コメント化を駆使してデバッグに役立てたり……と用途は様々あるんですよ？

```
#include <stdio.h>

int main(void){
    //出力する
    printf("Hello, world");
    //プログラムの終
    return 0;
}
```

Column) コメントアウト

コメントアウトには大きく分けて2種類あります。

1つ目は // です。その行の // のあとの文字は全てコメント扱いになります。

右における灰色の部分がコメントにあたります。プログラム実行時は、コメントは全て無視されるので、こんな感じでプログラムの説明を書くことができます。

```
#include <stdio.h>

int main(void){
    int a, b, c; //int型変数a, b, cを宣言
    a = 3; //aに3を代入
    b = 7; //bに7を代入
    c = a+b; //cにa+bを代入
    printf("aとbの和は%dです。", c); //出力
    return 0; //プログラムの終了
}
```

▲ 灰色部分がコメント

Column) コメントアウト

もう1つは、`/**/`です。`/*` から `*/` までで囲まれたすべての文字列は(行を跨いでいても)全てコメント扱いになります。

こちらは、まとめてコメントアウトするときに便利です。

今回の講座は以上です。そういえば、皆さんは早稲田の校歌は歌えますか??? 僕は無理です!!!

```
#include <stdio.h>

int main(void){
    /*aとbの和を出力するプログラム*/
    int a, b, c;
    a = 3;
    b = 7;
    c = a+b;
    /*都の西北 早稲田の森に
    聳ゆる薨 われらが母校*/
    printf("aとbの和は%dです。", c);
    return 0;
}
/*わせだ わせだ わせだ
           わせだ   わせだ
   わせだ   わせだ
*/
```

▲ 灰色部分がコメント

Question4-1) 偶数・奇数

以下のような挙動をするプログラムを作成せよ。ただし、実行結果は下記の【実行例】に準ずるようにせよ。

①

自然数を1つ入力する。

②

偶数を入力した場合は、「even」と出力する。
奇数を入力した場合は、「odd」と出力する。
いずれかを出力したら、プログラムを終了する。

【実行例1】

```
117 [Enter]  
odd
```

【実行例2】

```
2000 [Enter]  
even
```

Answer4-1) 偶数・奇数

```
#include <stdio.h>

int main(void){
    int num;
    scanf("%d", &num);
    if(num % 2 == 0) printf("even¥n");
    else printf("odd¥n");
    return 0;
}
```

▲ Program4.8 even_odd.c

解答例は、左のProgram4.8の通りです。偶数が「2で割ったときの余りが0である数」と気づければ簡単でしたね。

解答例では、if文を使用しましたが、switch文でも書けるので是非チャレンジしてみてください。

諸事情により、今回の演習問題はこの1問だけです！ごめんなさい！