

## ソフトゼミ A 第 3 回 解答と解説

### ■ 練習問題

1. a03\_2.c を改造して、入力された数が 3 の倍数であれば、「3 の倍数です」と、そうでなければ「3 の倍数ではありません」と出力するプログラムを作れ。

```
#include <stdio.h>
int main( void ){
    int n;
    scanf( "%d", &n );
    if( n % 3 == 0 ){
        printf( "3 の倍数です¥n" );
    }else{
        printf( "3 の倍数ではありません¥n" );
    }
    return 0;
}
```

$n$  の倍数である  $\Leftrightarrow n$  で割った余りが 0 である

です。よく使うので覚えておきましょう。

Sample Input

5

Sample Output

3 の倍数ではありません

2. 身長が 120[cm]未満の人は乗れない、130[cm]未満の人は保護者の同伴が必要なジェットコースターがある。身長[cm]を入力して、その人がジェットコースターに乗るための条件を満たしているかどうかを判定し、結果を画面に表示するプログラムを書け。保護者の同伴が必要な場合は、その旨を示せ。

```
#include <stdio.h>
int main( void ){
    double height;
    (次ページへ続く)
```

```

scanf( "%lf", &height );
if( height < 120.0 ){
    printf( "乗れません¥n" );
}else if( height < 130.0 ){
    printf( "乗れますが、保護者の同伴が必要です¥n" );
}else{
    printf( "乗れます¥n" );
}
return 0;
}

```

一応 double 型でやってみました。身長が 120.0[cm]以上の時は else if ( height < 130.0 ){~の部分が実行され、身長が 130.0[cm]以上の時は else~の部分が実行されます。else if の部分を実行している時、既に身長は 120[cm]を超えていることがわかっているので、else if( height >= 120.0 && height < 13.0)~のように、改めて 120[cm]以上であることを確かめる必要はありません。

#### Sample Input

156.0

#### Sample Output

乗れます

3. 整数を 3 つ入力させ、1 番大きい数を表示させるプログラムを作れ。

```

#include <stdio.h>
int main( void ){
    int a, b, c;
    int max;
    scanf( "%d%d%d", &a, &b, &c );
    max = a;
    if( max < b ){
        max = b;
    }
    if( max < c ){
        max = c;
    }
    (次ページへ続く)
}

```

```
}  
printf( "%d¥n", max );  
return 0;  
}
```

いろいろ方法はありますが、ここでは最大値を更新していく方法を取り上げます。まず、a を仮の最大値とします。次に、b が a より大きければ、b を仮の最大値とします。同様に、c と今までの最大値を比較してみて、最終的な最大値を決定します。

#### Sample Input

8192 65536 16384

#### Sample Output

65536

4. 西暦を入力し、その年の干支を教えてくれるプログラムを `switch` 文で作れ。

```
#include <stdio.h>  
int main( void ){  
    int year, eto;  
    scanf( "%d", &year );  
    eto = year % 12;  
    switch( eto ){  
        case 0:  
            printf( "さる¥n" );  
            break;  
        case 1:  
            printf( "とり¥n" );  
            break;  
        case 2:  
            printf( "いぬ¥n" );  
            break;  
        case 3:  
            printf( "いのしし¥n" );  
            break;  
        case 4:  
            printf( "うし¥n" );  
            break;  
        case 5:  
            printf( "えび¥n" );  
            break;  
        case 6:  
            printf( "とら¥n" );  
            break;  
        case 7:  
            printf( "うま¥n" );  
            break;  
        case 8:  
            printf( "おとこ¥n" );  
            break;  
        case 9:  
            printf( "めい¥n" );  
            break;  
        case 10:  
            printf( "いよめ¥n" );  
            break;  
        case 11:  
            printf( "子¥n" );  
            break;  
    }  
}
```

(次ページへ続く)

```

    printf("ねずみ¥n");
    break;
case 5:
    printf("うし¥n");
    break;
case 6:
    printf("とら¥n");
    break;
case 7:
    printf("うさぎ¥n");
    break;
case 8:
    printf("たつ¥n");
    break;
case 9:
    printf("へび¥n");
    break;
case 10:
    printf("うま¥n");
    break;
case 11:
    printf("ひつじ¥n");
    break;
}

    return 0;
}

```

長つw あ、**break** は絶対に忘れないでね！**break** がないと、2 つ以上の干支が出力されることがあります。(この場合の)**switch** 文の真の意味は、**eto** が 1 の時には「**case 1:**」と書かれているところの下にジャンプする、**eto** が 2 の時には「**case 2:**」と書かれているところの下にジャンプする...といったもので、**break** がないと、そのまま実行を続けてしまいます。途中で **break;**があると、その場で **switch** 文から抜けられます。

## ■ 追加練習問題

1. 数を4つ入力して、4つとも7の倍数なら「全部7の倍数です」、7の倍数が含まれているなら「7の倍数があります」、7の倍数が1つもないなら「7の倍数はありません」と出力するプログラムを作れ。

```
#include <stdio.h>

int main( void ){
    int a, b, c, d;
    int counter = 0;
    scanf( "%d%d%d%d", &a, &b, &c, &d );
    if( !( a % 7 ) ){ counter++; }
    if( !( b % 7 ) ){ counter++; }
    if( !( c % 7 ) ){ counter++; }
    if( !( d % 7 ) ){ counter++; }
    if( counter == 4 ){
        printf( "全部 7 の倍数です\n" );
    } else if( counter > 0 ){
        printf( "7 の倍数があります\n" );
    } else {
        printf( "7 の倍数はありません\n" );
    }

    return 0;
}
```

練習問題と同じように、 $a \% 7 == 0$ で判定してもいいのですが、同じなのもつまらないので $!(a \% 7)$ という条件式で判定しています。「!」演算子には、!の右側にある数字を

- その数字が0(偽)なら1(真)にする
- その数字が0以外(真)なら0(偽)にする

効果があります。if文は、括弧の中身が0以外の場合(真)の時のみに次の文(または{~}のいブロック)を実行しますから、 $!(a \% 7)$ で、 $a \% 7 == 0$ と同じ意味の条件式となります。

Sample Input

110 111 112 113

## Sample Output

7の倍数があります

2. 攻撃力  $a$  の勇者が、体力  $hp$ 、防御力  $d$  のモンスターに挑む。このモンスターを一撃で倒せるかどうかを判定せよ。ただし、ダメージ計算式は  
(ダメージ) = (攻撃側の攻撃力)  $\times$  2 - (防御側の防御力)  
であり、攻撃は必ず当たるものとする。  
入力の順番は  $a \rightarrow hp \rightarrow d$  である。

```
#include <stdio.h>
int main( void ){
    int a, hp, d;
    scanf("%d%d%d", &a, &hp, &d);
    if( hp <= a * 2 - d ){
        printf( "一撃で倒せます\n" );
    }else{
        printf( "一撃では倒せません\n" );
    }
    return 0;
}
```

不等号「 $\leq$ 」は「 $<=$ 」、「 $\geq$ 」は「 $>=$ 」と表します。「 $<$ (もしくは $>$ )」と「 $=$ 」の順番を間違えたり、「 $<$ (もしくは $>$ )」と「 $=$ 」の間にスペースを空けたりしないでください。

## Sample Input

100 200 300

## Sample Output

一撃では倒せません

3. 体力  $h1$ 、攻撃力  $a1$ 、防御力  $d1$  の勇者が、体力  $h2$ 、攻撃力  $a2$ 、防御力  $d2$  のモンスターに挑む。このモンスターを3ターン以内で倒せるかどうかを判定せよ。  
攻撃の順番は、勇者 $\rightarrow$ モンスター $\rightarrow$ 勇者 $\rightarrow$ モンスター $\rightarrow$ 勇者 $\rightarrow$ モンスターであり、ダメージ計算式は前回と同じものを用いよ。やはり攻撃は必ず当たるものとする。  
なお、勇者は体力が10以下になると攻撃力が3倍になるスキルを身に着けている。  
入力の順番は  $h1 \rightarrow a1 \rightarrow d1 \rightarrow h2 \rightarrow a2 \rightarrow d2$  である。

```

#include <stdio.h>

int main( void ){
    int h1, a1, a1x, d1;
    int h2, a2, d2;
    scanf( "%d%d%d%d%d%d", &h1, &a1, &d1, &h2, &a2, &d2 );
    a1x = a1 * 3;

    /* turn 1 */
    if( a1 <= 10 ){
        h2 = h2 - ( a1 * 2 - d2 );
    }else{
        h2 = h2 - ( a1x * 2 - d2 );
    }
    if( h2 <= 0 ){
        printf( "倒せませぬ\n" );
        return 0;
    }
    h1 = h1 - ( a2 * 2 - d1 );
    if( h1 <= 0 ){
        printf( "倒せませぬ\n" );
        return 0;
    }

    /* turn 2 */
    if( a1 <= 10 ){
        h2 = h2 - ( a1 * 2 - d2 );
    }else{
        h2 = h2 - ( a1x * 2 - d2 );
    }
    if( h2 <= 0 ){
        printf( "倒せませぬ\n" );
        return 0;
    }
}
(次ページへ続く)

```

```

h1 = h1 - ( a2 * 2 - d1 );
if( h1 <= 0 ){
    printf( "倒せません¥n" );
    return 0;
}

/* turn 3 */
if( a1 <= 10 ){
    h2 = h2 - ( a1 * 2 - d2 );
}else{
    h2 = h2 - ( a1x * 2 - d2 );
}
if( h2 <= 0 ){
    printf( "倒せます¥n" );
    return 0;
}else{
    printf( "倒せません¥n" );
}

return 0;
}

```

※/\* ~ \*/は「コメント」で実行ファイルに何の影響も及ぼしません。注釈などが必要と感じたら使ってください。

a1xには、プレイヤーがピンチの時に攻撃力が上がった時の攻撃力を格納しています。なお、途中でプレイヤーか敵かのどちらかのHPが0になった場合、「return 0;」で処理を打ち切っています。なお、この問題はfor文かwhile文を用いると格段に簡単に、そして短く書けます。for文で書いてみた時との比較も勉強になるかと思います。

#### Sample Input

9999 9999 9999 9999 9999 9999

#### Sample Output

倒せます