



# 知能情報工学演習I 第10回(後半第4回)

岩村雅一

[masa@cs.osakafu-u.ac.jp](mailto:masa@cs.osakafu-u.ac.jp)

# 前回の課題

## ■ 課題1

- 10進数の整数を入力し、8進数と16進数を入力するプログラムを作りなさい。

```
#include<stdio.h>

int main(void){
    int a;
    printf("10進数を入力: ");
    scanf("%d",&a);
    printf("8進数: %o\n",a);
    printf("16進数: %x\n",a);

    return 0;
}
```

# 前回の課題

## ■ 課題2

- 身長、体重、名前を入力し、名前とBMIを返すプログラムを作りなさい。
- 出力例:「○○さんのBMIは○○です。」

```
#include<stdio.h>

int main(void) {
    float h, w;
    char name[100];

    printf("名前: ");
    scanf("%s", name);
    printf("身長(m): ");
    scanf("%f", &h);
    printf("体重(kg): ");
    scanf("%f", &w);
    printf("%sさんのBMIは%fです。¥n",
           name, w/(h*h));

    return 0;
}
```

### ■ 課題3

- 整数を入力し、10の位を四捨五入するプログラムを作りなさい。
- 例: 123を入力すると、100を返す。789を入力すると、800を返す。

#### 回答例1: float型で入力

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

float  
int

```
int main(void) {  
    float a;  
    printf("Input: ");  
    scanf("%f",&a);  
    printf("%dの四捨五入は%d¥n",  
           (int)a, (int)(a/100.0+0.5)*100);
```

```
    return 0;
```

```
}
```

$123.0 / 100.0 + 0.5$   
 $\rightarrow 1.73$

$1.73 \rightarrow 1$

#### 回答例2: int型で入力

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

```
int main(void) {  
    int a;  
    printf("Input: ");  
    scanf("%d",&a);  
    printf("%dの四捨五入は%d¥n", a,  
           (a+50)/100*100);
```

```
    return 0;
```

```
}
```

$123 + 50 \rightarrow 173$

$173 / 100 \rightarrow 1$

# 実際にあった間違い

- 配列の範囲外を使用
  - `int a[1];`  
`scanf("%d",&a[1]);`
- `scanf`で文字列を入力するときに、&をつける
  - `char name[100];`  
`scanf("%s",&name);`
- 課題2で名前を入力させない
- 課題3で常に答えが0
- コンパイルできない
- 改行がないため、結果が表示されない

# 後半の予定

7. 6月1日 プログラミング環境(テキスト1,2章)
8. 6月8日 変数とデータ型(3章)、演算子(4章)
9. 6月15日 コンソール入出力(6章)、配列(3章)
10. 6月22日 制御文1(テキスト5章)
11. 6月29日 制御文2(テキスト5章)
12. 7月13日 関数1(テキスト7章)
13. 7月20日 配列(3章)、応用プログラム

# 本日のメニュー

## ■ 制御文

### □ 条件分岐

#### ■ if文

#### ■ switch文

### □ 繰り返し

#### ■ for文

#### ■ while文

#### ■ do-while文

# if文

- 条件を満たすときに命令を実行する

```
int a;  
a=10;  
if (a==10)
```

```
printf("aは10です。¥n ");
```

命令

- 条件の書き方(関係・等価演算子、テキストP.78)

- == 等しい

- != 等しくない

- > 大なり

- < 小なり

- >= 等号を含む大なり( $\geq$ )

- <= 等号を含む小なり( $\leq$ )

# if-else文

- 条件を満たすと命令Aを実行し、満たさないときは命令Bを実行する

```
int a;  
scanf("%d", &a);  
if (a==10)
```

```
printf("aは10です。¥n ");
```

```
else
```

```
printf("aは10ではありません。¥n ");
```

命令A

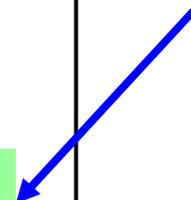
命令B

# 複合文 (ブロック)

- 2行以上の文を実行したいとき

```
int a;  
scanf("%d", &a);  
if (a==10) {  
    printf("aは10です。¥n ");  
    printf("へへーん。¥n ");  
} else {  
    printf("aは10ではありません。¥n ");  
    printf("ほほーん。¥n ");  
}
```

ブロック



# if文の入れ子

## ■ if文の中にif文

```
int a;  
scanf("%d", &a);  
if (a==10)  
    printf("aは10です。¥n ");  
else {  
    if (a==11)  
        printf("aは11です。¥n ");  
    else  
        printf("aは10でも11でもありません。¥n ");  
}
```

# if文のサンプルプログラム

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

```
int main(void) {
```

```
    int a;
```

```
    scanf("%d", &a);
```

```
    if (a==10) {
```

```
        printf("Bingo!!¥n");
```

```
    } else {
```

```
        if (a<10) {
```

```
            printf("もっと大きい。¥n");
```

```
        } else {
```

```
            printf("もっと小さい。¥n");
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

# switch文

```
int a;  
scanf("%d", &a);  
  
switch(a) {  
case 10:  
    printf("aは10です。¥n");  
    break;  
case 11:  
    printf("aは11です。¥n");  
    break;  
default:  
    printf("aは10でも11でもありません。¥n");  
}
```

# for文

## ■ 条件を満たす間繰り返す

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

```
int main(void) {
```

```
int i;
```

```
for (i=0; i<3; i++) {
```

```
    printf("i=%d\n", i);
```

```
}
```

```
return 0;
```

```
}
```

初期化式

条件式

再設定式(増分)

# for文

```
i=0 ↓  
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

```
i=0  
i=1  
i=2
```

# for文

i++を実行

i=0



i=1

```
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

i=0

i=1

i=2

# for文

条件のチェック → 条件を満たす

i=1

```
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

```
i=0  
i=1  
i=2
```

# for文

`i=1`

```
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

```
i=0  
i=1  
i=2
```

# for文

i++を実行

i=1



i=2

```
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

i=0

i=1

i=2

# for文

条件のチェック → 条件を満たす

i=2

```
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

```
i=0  
i=1  
i=2
```

# for文

`i=2`

```
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

```
i=0  
i=1  
i=2
```

# for文

i++を実行

i=2



i=3

```
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

i=0

i=1

i=2

# for文

条件のチェック → 条件を満たさないので  
for文のループを終了

i=3

```
for (i=0; i<3; i++) {  
    printf("i=%d¥n", i);  
}
```

## 出力

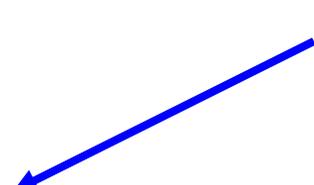
```
i=0  
i=1  
i=2
```

# while文

- 条件を満たす間繰り返す
  - 条件式のみ
  - 下の例はさっきのfor文と等価

```
int i;  
i=0;  
while (i<3) {  
    printf("i=%d¥n", i);  
    i++;  
}
```

条件式



# do-while文

- 条件を満たす間繰り返す
  - 基本的にwhile文と同じ
  - 条件判定は後
  - 下の例はさっきのfor文、while文と等価

```
int i;  
i=0;  
do {  
    printf("i=%d¥n", i); 条件式  
    i++;  
} while(i<3);
```

