

# C言語講座2015 第2回

ループ(while文,for文),分岐(if文,switch文)



# はじめに演算子を確認！

## \* 代入演算子

代入演算子	コード	意味(等価コード)
=	b=a;	aをbに代入
+=	b+=a;	b=b+a;
-=	b-=a;	b=b-a;
*=	b*=a;	b=b*a;
/=	b/=a;	b=b/a;
%=	b%=a;	b=b%a;

## \* 単項演算子

単項演算子	コード	意味(等価コード)
-	b=-a;	b=(-1)*a;
++	++b; b++;	b=b+1;
--	--b; b--;	b=b-1;

## \* 比較演算子

比較演算子(コード)	意味
a==b	aとbは等しい
a!=b	aはbと等しくない
a<b	aはbより小さい
a<=b	aはb以下
a>b	aはbより大きい
a>=b	aはb以上

演算結果が真なら1,偽なら0 (例)5<4 ならば 0

## \* 論理演算子

比較演算子	意味
!	否定
&&	かつ
	または

演算結果が真なら1,偽なら0

(例)

x=10のとき

$a = (2 \leq x) \&\& (x \leq 9)$       $1 \&\& 0 \rightarrow 0$

**※「 $2 < x < 9$ 」という書き方はNG !**

xが何であれ「 $2 < x$ 」は0か1。 $(0 \text{か} 1) < 9$ は必ず1になる。つまり,常に1になる!

# ループ構文

## \* while文

### while文の構造

```
while(条件式){  
    命令文  
}
```

条件式が真のときwhile文の中の命令を繰り返す。





(例)

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

```
int main(){
```

```
    int i=1,sum=0;
```

```
    while(i <=100){
```

```
        sum+=i;
```

```
        i++;
```

```
    }
```

```
    printf("%d\n",sum);
```

```
}
```

// i と sum を定義&初期化

// xが100以下の間ループ

// sum=sum+iと同じ。sumにiを加える

// iに1足す

// 1から100までの和を出力



## \* do while文

### do while文の構造

```
do{
```

命令文

```
} while(条件式); //セミicolon注意
```

(例) 先の例のmain文の中は...

```
int i=1,sum=0;
```

```
do{
```

```
    sum+=i;
```

```
    i++;
```

```
} while(i<=100);
```

```
printf("%d\n",sum);
```

となる。

[while文とdo while文の違い]

while文 : 前判定(次のループを行うか)  
do while文 : 後判定(今のループを続けるか)

## \* for文

### for文の構造

```
for(初期化式;条件式;更新式){  
    命令文  
}
```

初期化式を実行し、条件式が真の間更新式(カウンタ)に従ってループする。





(例)

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

```
int main(){
```

```
    int i,sum;
```

```
    for(i=1,sum=0;i<=100;i++){
```

```
        sum+=i;
```

```
    }
```

```
    printf("1から100までの和は¥"%d¥"¥n",sum);
```

```
}
```

初期状態は*i=1,sum=0*。

1 周目sumに+i する。(命令文)

ループしたら*i*に+1して2 周目に突入。

同じくsumに+i する。(命令文)

これを*i*が100になるまで繰り返す。。

*i* が100以下の間回る

「\」を表示するときは  
「¥」と書く

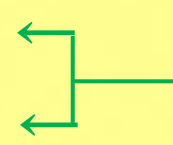
ここで！！

## <練習問題(^.^)/①>

int型変数a,bに値を読み込み(前回の復習), for文を使ってaからbまでの和を求めるプログラムを作しましょう！

例えばa=3,b=10としたら,  
実行結果は

「a= 3  
b= 1 0



3 と 1 0 はscanfで読み込む

3 から 1 0 までの和は “5 2” です。」

(青字を%dで出してみましよう！前回の復習！)

となるようにしてみてください。



# <練習問題(^.^)/①>の解答例

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

```
int main(){
```

```
    int a,b,i,sum;
```

```
    printf("a=");
```

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

```
    printf("b=");
```

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

```
    for(i=a,sum=0;i<=b;i++){    // i,sumの初期化, iがb以下の間ループさせる
```

```
        sum+=i;                // sumの初期値を0にしたのはということ
```

```
    }
```

```
    printf("%dから%dまでの和は¥"%d¥"¥n",a,b,sum);
```

```
}
```

# 分岐構文

## \* if文

### if文の構造

```
if(条件式 1 ){  
    命令文 1           //条件式 1 が真のとき実行  
}else if(条件式 2 ){  
    命令文 2           //条件式 1 が偽かつ条件式 2 が真のとき実行  
}  
    . . . . .  
else{  
    命令文             //上の条件のいずれにも当てはまらないとき実行  
}
```



(例)

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

```
int main(){
```

```
    printf("a=");
```

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

//aの値を読み込む

```
    if(a%3==0){
```

// 3 で割った余りが 0 なら 3 の倍数

```
        printf("%dは 3 の倍数です。¥n",a);
```

```
    }else
```

```
        printf("%dは 3 の倍数ではありません。¥n",a);
```

```
}
```





# \* switch文

## switch文の構造

```
switch(変数){  
case 定数式 1 :  
    命令文 1  
    break;  
case 定数式 2 :  
    命令文 2  
    break;  
  
    . . . . .  
  
default:  
    命令文  
}
```

break文はfor,while,do,switch文中で、ループを抜け出す役割

defaultは省略可

# (例)

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

```
int main(){
```

```
    int a;
```

```
    printf("a=");
```

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

```
    switch(a){
```

```
        case 100:
```

```
            printf("変数は100¥n");
```

```
            break;
```

```
        case 50:
```

```
            printf("変数は50¥n");
```

```
            break;
```

```
        default:
```

```
            printf("その他¥n");
```

```
    }
```

```
}
```

// 「;」ではなく「:」！注意！

// 「break;」を忘れずに！



# おまけ 1

## \* continue文

ループの処理のうち、その回の処理をパスする

(例)

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

```
int main(){
```

```
    int n;
```

```
    for(n=1;n<=3;n++){
```

```
        printf("%d start¥n",n);
```

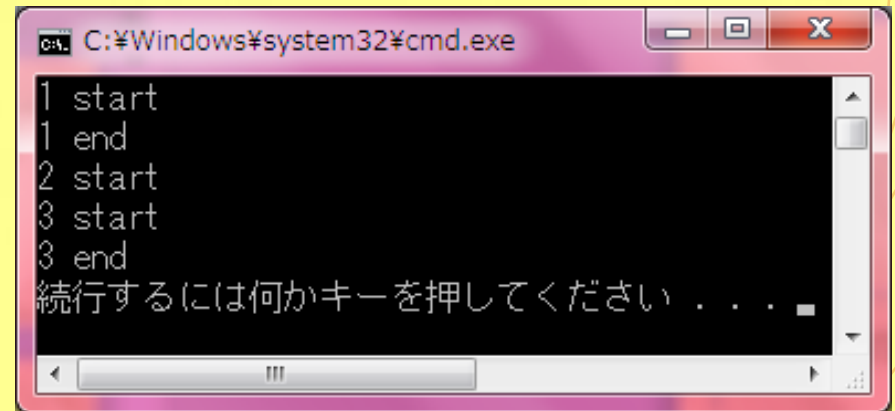
```
        if(n==2) continue;
```

```
        printf("%d end¥n",n);
```

```
    }
```

```
}
```

[実行結果]



```
C:\Windows\system32\cmd.exe
1 start
1 end
2 start
3 start
3 end
続行するには何かキーを押してください . . .
```

「2 end」の処理がパスされている。

# おまけ2

## \* 乱数の取得(先取り！)

乱数を出す関数(これを書くと乱数は返してくれる)

- rand()
- srand()

(例)

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

```
int main(){
    int x, i;
    for(i = 0; i < 10; i++){
        x = rand() % 6 + 1;
        printf("出た目は = %d\n", x);
    }
}
```

randを使うのに  
必須!!

など

rand()とは0から最大値を定数RAND\_MAXまでの値で返す関数。乱数を6で割った時の余りである0～5を出現させ+1をすることによって1～6の値を出現させた。

しかし、これでは同じ実行結果しかでない...



そこで！！

time関数を使うのに必須!!

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

//追加部分

srandはrandの初期設定をするようなもの

```
int main() {
    int x, i;
    srand((unsigned)time(NULL));
    for(i = 0; i < 10; i++){
        x = rand() % 6 + 1;
        printf("出た目は = %d\n",x);
    }
}
```

//追加部分

こうするとsrandには常に違った数値が渡されることになる。



それではラスト！＜練習問題②(・ω・)?＞

あるテストのスコアを乱数で出し(斬新だけど)、  
そのスコアに対して、

0～29点, 30～49点, 50～69点,  
70～89点, 90～100点

の5段階で評価してみましょう！

評価の仕方は自分の好きなようにやってみま  
しょう！また評価する前にスコアがいくつなの  
かをprintfで書いておきましょう！

ヒント：評価のところはif文！



# <練習問題②(・ω・)?>の解答例

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

int main(){
    int x;
    srand((unsigned)time(NULL));
    x = rand() % 100 + 1;
    printf("今回のあなたの点数は%d点\n",x);
    printf("これはこれは...\n");

    if(90<=x && x<=100){
        printf("チョーラッキー!!\n");
    }else if(70<=x){
        printf("すごいぞ！おめでとう！\n");
    }else if(50<=x){
        printf("まずまずですな。 \n");
    }else if(30<=x){
        printf("残念な結果です...\n");
    }else
        printf("なんて目だっ！！！！\n");
    }
}
```



今日はこれでおしまいです！  
お疲れ様でした！！