C言語講座　演習×３

分岐、ループについて。
その中で演算子、配列について

・・・内容が文字ばかりですみません

Ifを用いた分岐

「もしＡならば〇〇をする。

　もしＡではなくてＢならば△△をする。

　ＡでもなくＢでもなければ□□をする」

といったことをします。

さっそくですが具体例

```
char   k3  = ‘A’;

 if(   k3 == ‘A’  ){//もしAならば○○をする
        printf(“    ○○する     ”);

  }else if(   k3==‘B’ ){//もしAではなくてBならば△△をする
        printf(“    △△する     ”);

  }else{//AでもなくBでもなければ□□をする
        printf(“□□しちゃう”);
}
```

このような構造になっています

- If　　(条件式 1) {

　　条件式1が“真”ならば行う処理

　} else if(条件式２) {

　　条件式１が“偽”
　　条件式２が“真”の時行う命令

} else{
　　条件式のどれにも当てはまらない場合の命令
}

演算子

- 演算子とは、
  簡単にいうと記号を用いて演算を指示するもの。
  記号とか。

今回は算術演算子（四則演算で使ったやつ）を除いた比較演算子と論理演算子について。

If文などの中の条件式で使われます。

比較演算子

＝＝　（等しい）
！＝　（等しくない）
＜　　（より大きい）
＜＝　（～以上）
＞＝　（より小さい）
＞　　（未満）

「イコールは右
と覚えておくと迷わないかも

これらを用いて真偽の判別をします。
ただし論旨演算子を用いないで１つの式で２つ以上使うことができない。　ダメな例：　1< n <5

- If文と比較演算子を用いた簡単な例題

  次の（）内の処理は実行されるか。

- if（２＞＝４）{　命令　}

- if(３＊１０　==　（５－３）＊１５){　命令　}

- If文と比較演算子を用いた簡単な例題

次の（）内の処理は実行されるか。

if（２＞＝４）{　命令　}

→条件が偽である為、（）内の処理は行われない

- if(３＊１０　==　（５－３）＊１５){　命令　}

→条件が真である為、処理は実行される。

論理演算子

！（ＮＯＴ、否定）
＆＆（ＡＮＤ、かつ）
||（ＯＲ、または）

使い方　ｋ＝３　の場合

（（ｋ＝＝３）　＆＆　（ｋ＝＝４　））→偽
（（ｋ＝＝３）　｜｜　（ｋ＝＝４　））→真
（（ｋ＝＝３）　＆＆　!(ｋ＝＝４　））→真

では、演習問題へ

演習問題１（答えは次のページ）

０～１００の点数をキーボードから入力してもらい値によって次の評価を出力しなさい。

- 100点ちょうどならＡＡ
- ８５～９９点ならＡ
- ７０～８５点ならＢ
- ７０未満ならＣ
- o～１００以外の点数ならＤ

（同じような文はひたすらコピペするべし！）

解答例

```
#include<stdio.h>
int main(void){
    int n;
    printf(“点数を入力してください”);
    scanf(“%d”,&n);
    if(n<0 || n>100){
        printf(“D”);
    }else if{(n==100){
        printf(“AA”);
    }else if(n>=85 ){
         printf("A");
    }else if(n>=70){
        printf(“B”);
    }else {
        printf(“C”);
    }
}
```

switchを用いた分岐

```
switch(条件文){
case 条件１:
        命令１
        break;
case 条件２:
        命令２
        break;
default:
        命令３
        break;
}
```

← breakというループを　強制脱出させる命令文を使う。

注：breakは忘れないように。

default(条件外の処理)は省略ok

Whileを用いたループ

While（条件）　{
　　命令
}

仮に条件が真の時、
条件確認→{ }内の処理→条件確認→{}内の処理…
を繰り返します。逆に条件が偽であるなら{}を見られず実行もされません。

ループに入ってから条件が真のままだと限ループしてしまうので
主に次の２通りの方法でループの脱出をします。

{}内で条件値を変動させる

```
int n=1;

while(n<=5){

    printf(“%d週目”,n);

        n=n+1;
    }
```

breakというループを
強制脱出させる命令文を
使う。

```
int n =0;
while( 1 ){

if(n==5){
break;
}
n++;
}
```

コンピュータ上では
真＝１
偽＝０
という値でやりとりされています

do whileを用いたループ

```
do {
    命令
} while(条件式);
```

Whileとの違いは先に命令を処理してから条件を確認し、偽ならループせずに脱出、真なら命令を再び処理した後条件の確認....のループへ入ります。

Forを用いたループ

for(初期化式 ; 条件式 ; カウンタ){　命令　}

- 具体例↓

```
int k ;

for(k=1; k<10 ; k++ ){

printf(“%d回目のループ”,k);

}
```

k＝0から始まり、
条件k<5の確認、
printfの実行
カウンタ（k++される）
条件の確認……
というループ

どーっと詰め込んでしまいましたがとにかく手を動かして覚えましょう。

では、演習問題へ

演習２（答えは最後）

1から順に１００までコンソールに出力してください。
但し、その数が3で割り切れるならば数字の代わりにFizzと、5で割り切れるならBuzzと表示、3でも5でも割り切れる場合は、*FizzBuzz*と表示してください。

配列とは

今まではintやcharという宣言の単体であっちこっちからメモリを確保していました。

それを同一の型のデータをメモリ上に列にして連続で並べたものを配列といいます。

配列の使い方

簡単にいうと配列を使えばint型ならint型の変数を同時にいくつも作ることが出来ちゃいます。

例：

```
int  array[15];
```

array[0]からarray[14]までの１５個の変数が作成される。

→配列番号は０から作られます。

また、

配列の最後(今回はarray[15])ＮＵＬＬ(char型の場合￥０)が入ります。

配列のメリット

- int math[10];　....というように宣言した時に

math[n]やmath[3*n]

のように配列番号で管理できるので大量の変数を管理や編集が凄い楽になります！

配列の初期化

例：

- int c[3]={ 1 , 2 , 3 };
  　　または
  c[0]= 1 ;　 c[1]= 2 ;　 c[2]= 3 ;

- char n[4]= “ beat ” ;
  　　または

n[0]= ‘ b ’ ; n[1]= ‘ e ’ ; n[2]= ‘ a ’ ; n[3]=‘ t ’;

例：配列に入った数字を順,逆順に表示するプログラム

```
int main(){
    int i;
    int  n [5]={ 1 , 2 , 3 , 4 , 5 };
    for(i=0;i>=4;i++){
    printf(“%d ”, n [i]);
    }

 for(i=4;i<=0;i++){
    printf(“%d ”, n [i]);
    }
}
```

例：配列に入った数字を逆から入れなおすプログラム

```
int main(){
    int i,temp ,m;
    int  n [4]={ 1 , 2 , 3 , 4 };
    for(i=0;i<=1;i-++){
        n[i]=temp;
        n[i]=n[3-i];
        n[3-i]=temp;
    }
    for(m=0;m<4;m++){
        printf(“%d¥n”,n[i]);
    }
}
```

演習３（答えは最後おそらく宿題）

次の仕様の練習用の謎プログラムを作りなさい。

・初めにスイッチ文を用いて１か２をキーボードから入力してもらう。
　・１，２以外の時は再び入力させる
　・１と入力された時
もう一度数字を入力させdo while文を用いて合計値が１５を超えるまで入力させ続ける。入力のたびに「合計値は〇〇です」と出力。
（初めに１５以上の数字が入力された時の様子をwhile文に変えてみて確認みましょう。その違いは何か。）

　・２と入力された時はさらに４回数字を入力させ、
大きい順に並び替えたものを表示させてください。
（難しいです）

例: 12 ４ ５ 18 の順で入力　　→　18 12 5 4 の順番で出力

演習２の解答例

```
#include<stdio.h>
int main(){
    int i;
    for(i = 1; i <= 100; i++){
        if (i % 3 === 0 && i % 5 ===0){
            printf("FizzBuzz");
        }else if(i%3 === 0){
            printf("Fizz");
        }else if(i % 5 === 0){
            printf("Buzz");
        }else{
        printf(”%d”,i);
        }
}
```

演習３の解答例

```
#include<stdio.h>
int main(void){
    int n =0,m=0;
    int  temp;
    while( !(n==1) &&  !(n==2) ){
        scanf(“%d”,&n);
    }
    switch(n){
    case 1:
        do{
        scanf(“%d”,&n);
        m=m+n;
        printf(“合計値は%dです。”,m);
    } while(m<=15);
    break;
case 2:
int  array[4];
n=0;
m=0;
while(m<4){
    scanf(“%d”,&array[m]);
    m++;
}
for(m=0;m<3;m++){
    for(n=m+1;n<4;n++){
        if(array[n]>array[m]){
            temp=array[m];
            array[m]=array[n];
            array[n]=temp;
        }
     }
    }
For(m=0;m<4;m++){
printf(“%d”,array[m]);
}
break;
}}
```