

# 配列

## 前回の演習問題(その2)

4個の整数値を入力として受け取って、  
一番大きい値を返すプログラムを作成せよ。  
ただし、(その1)で作成した関数を3回呼び  
出すことによりプログラムを作成せよ。

## 簡単なプログラム

```
gets(buf);  
a=atoi(buf);  
gets(buf);  
b=atoi(buf);  
gets(buf);  
c=atoi(buf);  
gets(buf);  
d=atoi(buf);
```

めんどくさい

a, b, c, dを共通に扱えないか？

```
for (i=0; i<4;++i){  
  gets(buf);  
  a[i]=atoi(buf);  
}
```

for文を展開すると...

```
gets(buf);  
a[0]=atoi(buf);  
gets(buf);  
a[1]=atoi(buf);  
gets(buf);  
a[2]=atoi(buf);  
gets(buf);  
a[3]=atoi(buf);
```

実質的に、さきほどのと同じことをしている.

文法

最初の方で,

```
int a[4];
```

というように配列の宣言をする

これは,  $a[0]$ ,  $a[1]$ ,  $a[2]$ ,  $a[3]$ の4個を使いますという宣言.

配列の引数に変数を用いてもよい

例えば...

```
a[i]=atoi(buf);
```

はOK

```
a[i+2]=a[i+1]+a[i]
```

とかしてもOK(フィボナッチ数列)

# 小テスト(2回目)の解説

演習問題1の3まではできているとします

約数の個数を出力

前のプログラムをさらに改良して、約数の個数を出力するようにしてみよう.

解説によれば,

個数を数える時には, そのための変数(ここでは  $n$ )を用意して,

初期値ゼロを与えておく. あとは, 条件が成立した場合だけ  $n = n + 1$

(これは  $n++$  と書いても同じ) を行なうことで, ループが終了した時点での  
個数が  $n$  に残る.

```
int main(void){
    char buf[80]; int j; int i; int k; int n;
    gets(buf);
    j = atoi(buf);
    n = 0;
    for(i=1; i<=j; i++){
        k = j % i;
        if(k == 0){ n = n + 1; }
    }
    printf("%d¥n", n);
}
```

約数の和を求めたいので、  
if (k==0){n=n+i;}  
とする.

この数が、 $2*j$ の場合は  
出力するようにすればOK