

# 情報研究会CACTUS

## 第9回 講義資料

# 前回の練習問題の解答

## 練習問題 1

```
1 #include<stdio.h>
2 void cac(int a,int b,int *c,int *d,int *e,double *f,int *g){
3     *c=a+b;
4     *d=a-b;
5     *e=a*b;
6     *f=(double)a/b;
7     *g=a%b;
8 }
9 int main(){
10     int a,b,c,d,e,g;
11     double f;
12     scanf("%d",&a);
13     scanf("%d",&b);
14     cac(a,b,&c,&d,&e,&f,&g);
15     printf("%d,%d,%d,%.2f,%d",c,d,e,f,g);
16 }[EOF]
```

## 練習問題 2

```
1 #include<stdio.h>
2 #define N 50
3
4 void FibCalc(int *p); //フィボナッチ数列を計算する関数
5
6 int main(void){
7     int fib[N]={1,1}; //フィボナッチ数列の配列を用意し、先頭に1を入れておく
8     int n;
9     int *p=fib; //配列の先頭アドレスをポインタに格納する
10    FibCalc(p);
11    while(1){
12        printf("フィボナッチ数列の何項目を出力しますか? :%n");
13        scanf("%d",&n);
14        if(n<=0){
15            printf("不正な入力%n");
16        }else{
17            printf("フィボナッチ数列の%d項目:%d%n%n",n,fib[n-1]);
18        }
19    }
20    return 0;
21 }
22 void FibCalc(int *p){
23     int i;
24     for(i=2;i<N-2;i++){
25         p[i]=p[i-1]+p[i-2]; /*(p+i)=*(p+(i-1))+*(p+(i-2))
26         //フィボナッチ数列の公式
27     }
28 } [EOF]
```

今週の内容

文字列

# サンプルプログラム 1

```
1 #include<stdio.h>
2 int main(void){
3     char a[10];
4     char b[]="CACTUS";
5     char c[10];
6
7     a[0]='C';
8     a[1]='A';
9     a[2]='C';
10    a[3]='T';
11    a[4]='U';
12    a[5]='S';
13    a[6]='\0';//ヌル文字（ここ重要）
14    printf("%s\n",a);//一文字ずつ代入した場合
15
16    printf("%s\n",b);//最初から宣言（初期化）することも可能
17
18    printf("文字を入力してください");
19    scanf("%s",c);//scanfで入力するとき&はいらない
20    printf("%s",c);
21    return 0;
22 }[EOF]
```

実行結果

CACTUS  
CACTUS  
文字を入力してくださいcactus  
cactus

文字コード

文字コードとはコンピューターで使用する文字に番号(コード)を割り振ったもの

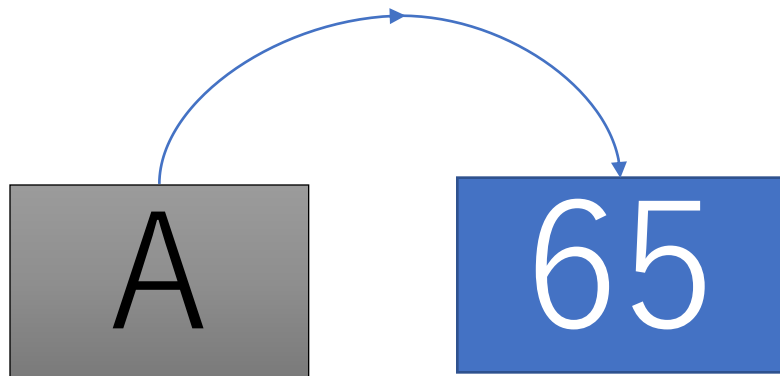
コンピュータの中では右の文字コードとして文字が記憶されている。

(例)

'A'=65

[illegible]

文字コードを参照することによってコンピュータは文字を認識することができる。



コンピュータ内で数値として記憶するという事は配列として定義すると文章を記憶できる



文字列

# 文字列のポイント

- 二重引用符“”で囲んだものを文字列リテラルと呼ぶ。  
printf(“ABC”);
- この文字列リテラルはヌル文字が末尾についた状態で記憶領域に格納される。

“ABC” → 記憶領域

A	B	C	¥0
---	---	---	----


ヌル文字とは値が0である文字のこと。  
8進拡張表記では‘¥0’と表記する。



文字列はchar(キャラ)型で宣言する。

文字をひとつずつ記憶するため配列のときと同じように宣言する。

(例)

char str[100];ヌル文字をふくむ100文字を格納できる  
char str[]={'A','B','C','¥0'};

入出力するときは%s(%cもあるが一文字しか表示できない)

scanfで入力するときは&がいない

# サンプル 1 の解説

宣言時に初期化

一文字ずつ代入

出力は % s

入力には & いらな

```
1 #include<stdio.h>
2 int main(void){
3     char a[10];
4     char b[]="CACTUS";
5     char c[10];
6
7     a[0]='C';
8     a[1]='A';
9     a[2]='C';
10    a[3]='T';
11    a[4]='U';
12    a[5]='S';
13    a[6]='\0';//ヌル文字 (ここ重要)
14    printf("%s\n",a);//一文字ずつ代入した場合
15
16    printf("%s\n",b);//最初から宣言 (初期化) することも可能
17
18    printf("文字を入力してください");
19    scanf("%s",c);//scanfで入力するとき&はいらな
20    printf("%s",c);
21    return 0;
22 }[EOF]
```

## サンプルコード 2

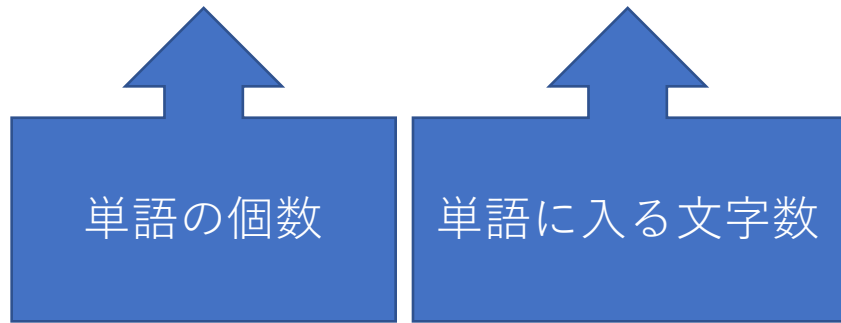
```
1 #include<stdio.h>
2 int main(void){
3     char str[3][10]={"abc","def","ghi"};
4     int i;
5
6     for(i=0;i<3;i++){
7         printf("%d-%s\n",i+1,str[i]);
8     }
9     return 0;
10 }[EOF]
```

実行結果

1 abc  
2 def  
3 ghi

## 文字列の 2 次元配列

```
char str[3][10] = {"abc", "def", "ghi"}
```



```
str[0] = "abc"
```

```
str[1] = "def"
```

```
str[2] = "ghi"
```

# 文字列を扱うライブラリ関数

```
1 #include<stdio.h>
2 #include<string.h>
3 int main(void){
4     char a[]="ant";
5     char b[]="bird";
6     char c[]="cat";
7     char d[]="dog";
8     char x[10];
9
10    //文字列の長さを調べる:strlen
11    printf("a[]の長さは%d\n",strlen(a));|
12
13    //xにbをコピーする:strcpy
14    printf("%s\n",strcpy(x,b));
15
```

```
16    //二つの文字を連結する:strcat
17    printf("%s\n",strcat(a,b));
18
19    //二つの文字を辞書順で比較する:strcmp
20    //c=d→0・c>d→1・c<d→-1
21    switch(strcmp(c,d)){
22        case 0:printf("二つは同じ\n");
23                break;
24        case 1:printf("%sが大きい\n",c);
25                break;
26        case -1:printf("%sが大きい\n",d);
27                break;
28    }
29 }
```

これは見るだけでいいよ

# 練習問題 1

キーボードから文字列を読み込み、その文字の 2 文字目を出力するプログラムを作ってみよう

## 練習問題 2

入力した文字の長さを調べるプログラムを作ってみよう。  
ただしstrlenは使用禁止

ヒント：文字列の最後にはヌル文字がくることよ

# 発展問題(やりたい人はどうぞ)

数字と文字が混ざった文字列を入力して数字を取り除いた文字列として表示してみよう

入力 a2b3c4d5

出力 abcd

(電電2年プログラミング課題)