

C言語講座Vol.3

2009年5月29日 CISC

字下げ

- 見やすく書こう！

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

```
int main(void)
{
    int a;
    printf("値=>");
    scanf("%d", &a);
    if(a>10)
    {
        printf("10より大きい値です¥n");
    }
    else
    {
        printf("10以下の値です¥n");
    }
    return 0;
}
```

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

```
int main(void)
{
    int a;
    printf("値=>");
    scanf("%d", &a);
    if(a>10)
    {
        printf("10より大きい値です¥n");
    }
    else
    {
        printf("10以下の値です¥n");
    }
    return 0;
}
```

別種の表記

<code>i = i + 1</code>	<code>i ++</code>	<code>++ i</code>
<code>i = i - 1</code>	<code>i --</code>	<code>-- i</code>
<code>i = i + 2</code>	<code>i += 2</code>	
<code>i = i - 2</code>	<code>i -= 2</code>	
<code>i = i * 2</code>	<code>i *= 2</code>	
<code>i = i / 2</code>	<code>i /= 2</code>	

```
printf ("同じです");  
puts ("同じです");  
putchar (' A');
```

```
scanf ("%c", &a);  
a=getchar ();
```

一部まだ教えていないことがあるので、
下の入力編は参考程度に。

++iとi++の違い

- 下記のプログラムの、「++i」を「i++」に書き換えて試してみよう。

```
#include <stdio.h>

int main()
{
    int i;
    i=0;
    if(++i == 0)
    {
        printf("評価時点で0¥n");
    }
    else
    {
        printf("評価時点で0以外¥n");
    }
    return 0;
}
```

キャスト

- 強制的に式の型を変換する方法。

```
int a, b;  
float c, d;
```

```
a=10; b=3;
```

```
c=a/b;
```

```
d=(float)a/b; //変換したい場所の前に、括弧で型名を囲って記述する。
```

配列

- 多数の変数を連続して扱う

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

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]

```
int main(void)
{
    int a[10], i;

    for (i=0; i<10; i++)
    {
        printf("a[%d]=? ¥n=>", i);
        scanf("%d", &a[i]);
    }

    for (i=0; i<10; i++)
    {
        printf("a[%d]=%d¥n", i, a[i]);
    }

    return 0;
}
```

変数名	アドレス	データ
num	0x0000	12
i	0x0002	2
a[0]	0x0004	123
a[1]	0x0006	987
a[2]	0x0008	456
a[3]	0x000A	31
cnt	0x000C	12

メモリ空間を
連続的に確保

文字

- C言語では、文字はそれぞれに番号を振り、その番号を変数で保持することで、文字を表している。

たとえば大文字のA

これは、「A」とシングルクォートで囲って書くことで、割り振られた番号を示すことができる。

これをchar型の変数で保持することで、いわゆる文字となる。

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

```
int main()  
{  
    printf("Aの番号は%d。16進数だと%X\n", 'A', 'A');  
    return 0;  
}
```

直接値を書くことも可能

文字コード

上位3ビット→ ↓下位4ビット	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAC	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF/NL	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

一文字だけ入力

- 次の命令、getchar()を利用することで、一文字だけ入力することができる。
- ほかに、scanfの場合は、%dではなく%cとすると、一文字だけ入力できる。

```
#include <stdio.h>

int main()
{
    char moji1, moji2;

    moji1 = getchar();    //一文字だけ入力
    scanf("%c", &moji2); //同上

    printf("%c %c\n", moji1, moji2); //出力も%c
    return 0;
}
```

文字列

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
c	h	i	b	a	-	u	\0

↓

chiba-u

末端文字
(バックslashまたは円マーク)

```
char a[8]="chiba-u";
```

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

```
int main()  
{
```

```
    char str[100];  
    printf("文字を入力=>");  
    scanf("%s", &str[0]);  
    //scanf("%s", str);  
    printf("%s\n", str);  
    return 0;
```

```
}
```

//と書いてもよい。
//&はいらない!

文字列操作関数

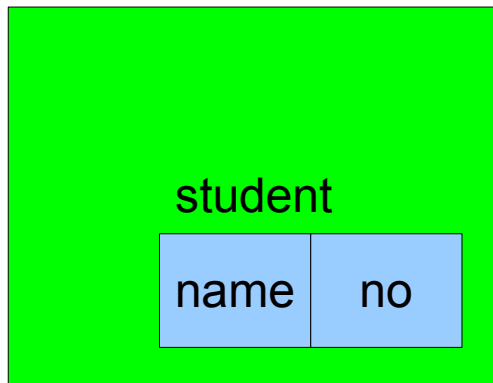
関数名	使い方	解説
strcpy	strcpy(文字列1, 文字列2);	文字列1に対し、文字列2をコピーする。
strncpy	strncpy(文字列1, 文字列2, 文字数);	文字列1に対し、文字列2を文字数分コピーする。
strcat	strcat(文字列1, 文字列2,);	文字列1の後ろに、文字列2をコピーする。
strncat	strncat(文字列1, 文字列2, 文字数);	文字列1の後ろに、文字列2を文字数分コピーする。
strcmp	返回值=strcmp(文字列1, 文字列2);	文字列1と文字列2を比較し、 文字列1>文字列2 なら正の値 文字列1=文字列2 なら0 を返す。
strlen	返回值=strlen(文字列);	与えられた文字列の終端文字を含まない長さを返す

記述例

```
char ccsName, str[100]="CCS";  
int flg;  
. . .  
strcpy(ccsName, str);  
flg=strcmp(ccsName, "CSS");
```

構造体

- 複数の変数をまとめて扱う



```
struct student{  
    char name[30];  
    int no;  
};
```

```
#include <stdio.h>  
  
struct student{  
    char name[30];  
    int no;  
};  
  
int main(void)  
{  
    struct student data[10];  
    int i;  
  
    for (i=0; i<10; i++)  
    {  
        data[i].no=i+1;  
        printf("名前を入力¥n=>");  
        scanf("%s", &(data[i].name[0]));  
    }  
    for (i=0; i<10; i++)  
    {  
        printf("No:%d¥n名前:%s¥n", data[i].no, data[i].name);  
    }  
    return 0;  
}
```

変数の初期値

- 変数を宣言した瞬間、初期値を入れる事ができる。

```
int a=0; //=で書いてやる。
int data[5]={10, 20, 12, 43, 165}, //このように、配列も初期値を入れられる。
    data2[]={10, 20, 12, 43, 165}; //要素数を指定しないと、適切な要素数を勝手に確保する！
char name[]="CISC"; //文字列もこうやって初期設定可能。
struct student{
    char name[100];
    int age;
} ccsmember1={"CISC", 19}; //構造体もこうやる。
```

課題1

- n 個の値 x ($0 < n \leq 10$)を配列に入力し、その合計、平均、最大値、最小値を求めるプログラムを作れ。
 - 0が入力されたり、入力された数が10個を超えたら、入力工程を終え計算を行う。
 - 入力のループと、計算のループは別にすること。
 - 最大値最小値を求めるアルゴリズムは次のとおり。
 - 最大値(最小値)を保持する変数を設定。
 - この変数と調べたい変数を比較し、最大値(最小値)の候補になりうるかを調べる。
(最大値(最小値)の変数の初期値はどうすればよい?)
- さらに、入力された値を昇順に整列して出力せよ。
(任意課題)

課題2

- 文字列(英小文字限定、100文字以内と自主規制)を入力し、もっとも使用されているアルファベットを調べて、出力するプログラムを作れ。
- ヒント
 - カウント用の配列(要素数は26個)を用意して、それぞれのアルファベットの数をカウントする。最後にその配列に対し最大値を求め、対応する要素のアルファベットを出力する。
 - ループ中のカウント自体はたった1行で可能。
 - 文字コードは連続している。さらに、'a'-'a'=?

課題3

- 任意課題
- 学籍番号、名前、性別をメンバに持つ構造体を定義し、次のデータを入力する。その後、学籍番号でソートして表示するプログラムを作れ。

学籍番号	名前	性別
09T001	千葉 太郎	男
09T043	千葉 花子	女
09T112	工学 太郎	男
09T080	福沢 漱石	男
08T077	夏目 諭吉	男

課題3のヒント1

- strcmpを使う？（以下使わない方法）
- 文字と数字の変換
 - 今現在われわれが利用しているC言語の環境上で扱う文字コードは、すくなくとも、'0'から'9'まで連続した値が割り当てられている。
→一桁なら、引き算一発で変換できる。
 - 二桁以上の数も、実は単純な式で変換できる。
上位の桁から(=左側の桁から)順に変換していった、そのたびに10倍してやると…。

課題3のヒント2

- 並べ替え(ソート)
今回の講座ではまだ一言も触れていないが、プログラミングの勉強で、ソートとは非常に基本的な、演習のひとつである。

今回は、各自が考えた方法や既に知っている方法を使ってよい。

まったく考え付かない場合は、次項参照。

課題3のヒント3

- 選択ソート
未整列の範囲内から、もっとも小さい(大きい)値を見つけ、未整列範囲先頭の要素と交換する。すると、未整列の範囲がひとつ狭まる。これを最後まで繰り返すと、全体が整列する。
- バブルソート
1番目と2番目を比較し、順番が逆であれば入れ換える。次に2番目と3番目を比較して入れ換える。これを最後まで行くと、最後の数だけが最小または最大の数として確定するので、確定していない部分について1つずつ減らしながら繰り返す。
(某所からコピー)

発展課題 リスト構造

- 配列は、複数のデータを扱えるが、すでに整列している状態下では、中間に新しいデータを挿入することが難しい。
そこで、リスト構造という概念を導入する。
- 詳しくは、進度に大きな差があった場合に、ホワイトボードにて説明する。

struct data
int nextNum;
char str[100]

data[0]	data[1]	data[2]	data[3]	data[4]
4	0	3	1	2
CICS	ほへい	フェイス	あさげ	ねいむ

課題1 解答

```
#include <stdio.h>

int main()
{
    int n, x[10], i, sum=0, max=0, min=10000;
    //入力
    for (i=0; i<10; i++)
    {
        do{
            printf("%d. 値を入力=>", i+1);
            scanf("%d", &x[i]);
        } while( x[i]<0 || 10000<=x[i]);
        if(x[i]==0)
        {
            break;
        }
    }
    n=i; //個数を代入
    //計算
    for (i=0; i<n; i++)
    {
        sum+=x[i];
        if(x[i]>max)
        {
            max=x[i];
        }
        if(x[i]<min)
        {
            min=x[i];
        }
    }
    printf("合計%d 最大値%d 最小値%d\n", sum, max, min);
    printf("平均%f\n", (float) sum/n);
    return 0;
}
```

課題2の解答1

```
#include <stdio.h>

int main()
{
    char str[100];
    int cnt[26];
    int i, flg=0, maxNum=0, maxChar;

    printf("文字列を入力\n=>");
    scanf("%s", str);

    //カウント用配列の初期化
    for (i=0; i<26; i++)
    {
        cnt[i]=0;
    }
    for (i=0; i<100 && str[i]!='\0'; i++)
    {
        //誤処理防止用
        if ('a' <= str[i] && str[i] <= 'z')
        {
            cnt[str[i]-'a']++;
        }
        else
        {
            printf("不正な文字が発見されました。 \n");
            flg=1;
            break;
        }
    }
}
```

課題2の解答2

```
for (i=0; i<26; i++)
{
    if (maxNum<cnt[i])
    {
        maxNum=cnt[i];
        maxChar=i+'a';
    }
}
printf("最も使われているアルファベットは、¥n");
printf("%c 個数:%d¥n", maxChar, maxNum);

return 0;
}
```