第 7 章　暗号を解読しよう

概要

プログラムで扱うデータには、数値以外に文字のデータを取り扱います。そこで文字を操作するための種々の関数について学びます。

この章の目標

文字の検索や文字数を数える関数を覚える。
英小文字を大文字に変換する関数を覚える。
数値と数字文字列の違いを理解する。

7.1　文字数を数えてみよう

---

11 章　文字を記録しよう　－文字変換－[pp.135-143]

●文字変数の宣言
既に第 2 章で下記のような文字変数の宣言を習いました。

(P 例 : 7.1-1)
```
#include<stdio.h>

main()
{
        char a;

        a = 'x';
        printf("a = %c¥n",a);
}
```

しかし、この変数 a は、1 文字しか代入できない変数でした。そこで、複数の文字（文字列）を代入できる配列を用いた宣言方法について学びます。
下記のプログラムは、キーボードから 9 文字、配列変数 b に代入する例です。

(P 例 : 7.1-2)
```
#include<stdio.h>

main()
{
        char b[10];

        scanf("%s",b);
        printf("b = %s¥n",b);
}
```

上記のプログラムを実行し、キーボードから hello と入力した場合、h からはじまる各文字は、配列にどのよう

に代入されるのでしょうか？
次のように、各文字は代入されます。

```
b[0] = 'h';
b[1] = 'e';
b[2] = 'l';
b[3] = 'l';
b[4] = 'o';
b[5] = 0;
```

ここで、一番最後の b[5]=0; とは、何を意味するのでしょうか？これは、文字列の終わりを表す終端文字です。この終端文字がないと、どこが文字列の終わりか分らないので、最後に終端文字をつけます。数値では、0 が終端文字を表します。

●文字列の色々な処理関数
終端文字（'¥0' or 0）で終わっている配列で表される文字列を対象として、文字数を数えたりする処理関数等を下記に示します。
なお、この処理関数を使用するためには、プログラムのはじめに下記のインクルードを記述しておく必要があります。

・include ファイル
```
#include<string.h>
```

・処理関数の一覧
```
size_t   strlen(string) : 文字数を数える。
                          括弧の中は配列名 (pointer)
                          '¥0'は数えない。size_t は、符号無し整数型
char     *strcpy(to_string,string) : 文字列の copy
                          最後の '¥0' も copy する。
char     *strncpy(to_string,string,length) : n 文字(length 文字数)の copy
char     *strcat(to_string,string) : 文字列の連結
                          to_string の最後の '¥0'に上書きして連結する。
char     *strchr(string,character) : 文字の検索
                          文字列の頭から検索し、有ればその pointer を無ければ NULL を返す。
int              strncmp( const char *string1, const char *string2, size_t count );
                          2 つの文字列の文字を比較します。n 文字の比較。
                          正:string1<string2, 0:等しい, 負:string1>string2
int              strcmp( const char *string1, const char *string2 );
                          2 つの文字列の文字を比較します。
                          正:string1<string2, 0:等しい, 負:string1>string2
```

●プログラム例
（P 例 : 7.1-3）
```
#include<stdio.h>
#include<string.h>

main()
{
        char a[]="information";
        char b[40];
```

```
        int     length;

        printf("original:%s¥n",a);

        strcpy(b,a);
        printf("copy:%s¥n",b);

        strcat(b,"+logistics");
        printf("add:%s¥n",b);

        length = strlen(b);
        printf("length=%d¥n",length);
}
```

・実行結果

```
original:information
copy:information
add:information+logistics
length=21
```

7.2　小文字から大文字に変化してみよう

●小文字と大文字

小文字から大文字へ、また、その逆の変換を行える関数を C 言語は持っています。

・include ファイル

```
#include<ctype.h>
```

・関数一覧

```
int     toupper(character) : 英小文字を大文字に変える。
int     tolower(character) : 英大小文字を小文字に変える。

int     character : 関数も引数も data 型は整数
```

●プログラム例

(P 例 : 7.2-1)

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

main()
{
        char a[100],b[100],c[100];
        int     i;

        printf("文字を入力 : ");
        gets(a);
        for(i=0; a[i] != 0; i++) {
                                        /* 終端文字が現れるまで繰り返し */
                b[i]=toupper(a[i]);
```

```
        }
        b[i]=0;                         /* 終端文字の末尾への付加 */

        for(i=0; a[i] != 0; i++) {
                c[i]=tolower(a[i]);
        }
        c[i]=0;

        printf("入力文字   :%s¥n",a);
        printf("大文字変換:%s¥n",b);
        printf("小文字変換:%s¥n",c);
}
```

●数値と数字文字列
文字として入力された数字を、int 型や double 型の数値に変換することができます。

・include ファイル
```
#include<stdlib.h>
```

・関数一覧
```
int      atoi(string) : 数字文字列を、int の数値に変換する。
                                (ascii to integer の意味)
double   atof(string) : 数字文字列を、double の数値に変換する。
```

●プログラム例
(P 例 : 7.2-2)
```
#include<stdio.h>
#include<stdlib.h>

main()
{
        char a[40];
        int     b;

        printf("数字を文字として入力 : ");
        gets(a);
        b=atoi(a);
        printf("文字:%s¥n",a);
        printf("数字:%d¥n",b);
}
```

7.3　文字に 1 を足して暗号化しよう

●文字の大きさ
計算機の内部では、文字は数値として表現されています。従って、次の例のような計算を行うことができます。

(P 例 : 7.3-1)

```
#include<stdio.h>

main()
{
        int i;
        char a,z;

        a = 'A';
        z = 'Z';

        for(i=0;i<26;i++){
                printf("a:%c, z:%c¥n",a,z);
                ++a;
                --z;
        }
}
```

●暗号を解こう

次のプログラムを実行した結果、 B!lbqqz!Ofx!Zfbs という結果を得ました。この暗号を解いて、下記のプログラムの ？？？ に入る文字列を求めて下さい。

(P 例 : 7.3-2)

```
#include<stdio.h>

main()
{
        int i;
        char a[] = "？？？";

        i=0;
        while(a[i] != 0){
                a[i] = a[i] + 1;
                ++i;
        }
        printf("%s¥n",a);
}
```

・実行結果

```
B!lbqqz!Ofx!Zfbs
```

●補足

例えば、"流通"という単語は、2 文字ですが、計算機では全角 1 文字を 2 文字として数えますので、この場合、4 文字となります。

実際に 7.1 で習った文字数を数える関数を使用して、文字数を数えてみましょう。
