

# プログラミング1

## Report#06

提出日: 2009/07/09(木)

所属: 工学部情報工学科

学籍番号: 095736E

氏名: 玉城 翔

問1.コマンドラインから受け取った文字列の大文字と小文字を変換するプログラムを作成せよ。入力は1バイトの表示文字とし、アルファベット文字以外は変換しない。

ソースプログラム:[parameter1.c]

```
/*
Program : parameter1.c
Comment : 大文字と小文字を変換するプログラム
*/

#include <stdio.h>
#include <ctype.h>

void replace(char *, char *); /*replace 関数(ポインタを引数にとる)*/

int main(int argc, char **argv){ /*main 関数(引数をとる)*/

    int i;
    char box[128];

    printf("sentence = %d\n",argc);

    for(i = 1, argv++; *argv != NULL; i++, argv++){

        printf("parameter(%2d)\t%s\n",i,*argv);

        replace(box, *argv); /*replace 関数の動作結果を与える*/

        printf("[%s] => [%s]\n",*argv,box); /*大文字・小文字変換を出力*/
    }

    return(0);
}

void replace(char *dest, char *str){ /*replace 関数の動作*/

    for( ; *str != 0; dest++, str++){

        if(islower(*str)){ /*小文字ならば*/
            *dest = toupper(*str); /*大文字変換*/
        }else if(isupper(*str)){ /*大文字ならば*/
            *dest = tolower(*str); /*小文字変換*/
        }else {
            *dest = *str; /*それ以外はそのまま*/
        }
    }
}
```

```
}
*dest = 0;
}
```

## 出力結果

```
[nw0936:~/prog1/para] e095736% ./parameter1 My NAME is Kakeru tamaSHIro

sentence = 6

parameter( 1) My
[My] => [mY]

parameter( 2) NAME
[NAME] => [name]

parameter( 3) is
[is] => [IS]

parameter( 4) Kakeru
[Kakeru] => [kAKERU]

parameter( 5) tamaSHIro
[tamaSHIro] => [TAMASHIRO]
```

## 考察

main 関数に引数をとる事により、コマンドラインのパラメーターの情報が出力することが分かる。  
int main(int argc, char \*\*argv)の中で、

argc : プログラム名とパラメーターの個数の和

\*argv : プログラム名とパラメーターを指すポインタ

```
コマンドライン:e095736% ./parameter1 My NAME is Kakeru tamaSHIro
argc = 6
```

*argv	parameter	My	NAME	is	Kakeru	tamaSHIro
	[0]	[1]	[2]	[3]	[4]	[5]

replace(char \*dest, char \*str) 関数で、引数をポインタでとることによって戻り値がなくても  
main 関数側で replace 関数が動作することが分かる。

## 問2.文字列を反転して表示するプログラムを作成せよ。

(例 "abcd" => "dcba")

ソースプログラム:[parameter2.c]

```
/*
Program : parameter2.c
```

```

Comment : 文字列を反転するプログラム
*/

#include <stdio.h>
#include <ctype.h>

int get_n(char *); /*get_n 関数(ポインタを引数にとる)*/
void revers(char *, char *, int); /*revers 関数(ポインタを引数にとる)*/

int main(int argc, char **argv){

    int i, word;
    char box[128];

    printf("sentence = %d\n",argc);

    for(i = 1, argv++; *argv != NULL; i++, argv++){

        printf("parameter(%2d)\t%s\n",i,*argv);

        word = get_n(*argv); /*get_n 関数の動作結果を与える*/
        revers(box, *argv, word); /*revers 関数の動作結果を与える*/

        printf("[%s] => [%s]\n",*argv,box); /*文字反転変換を出力*/
    }

    return(0);
}

int get_n(char *pa){ /*コマンドラインの文字数*/

    int i;

    for (i=0; *pa != 0; i++,pa++);

    return(i);
}

void revers(char *dest, char *str, int l){ /*文字反転変換を行う*/

    while( --l > 0) str++;
    for( ; *str != 0; dest++,str-- )
        *dest = *str;
    *dest = 0;
}

```

## 出力結果

```

[nw0936:~/prog1/para] e095736% ./parameter2 yM emaN sl urekaK orihsamaT

sentence = 6

```

```

parameter( 1) yM
[yM] => [My]

parameter( 2) emaN
[emaN] => [Name]

parameter( 3) sl
[sl] => [ls]

parameter( 4) urekaK
[urekaK] => [Kakeru]

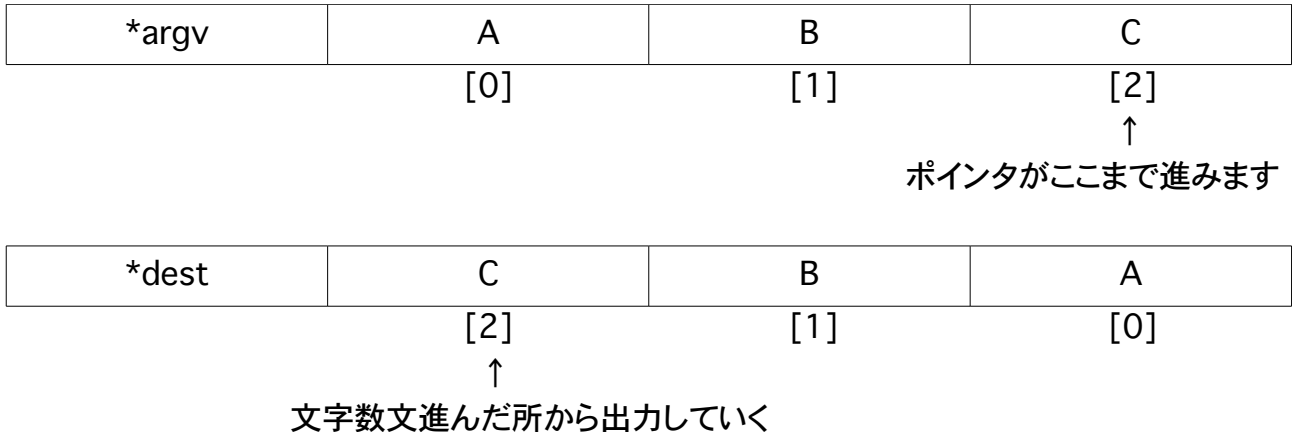
parameter( 5) orihsamaT
[orihsamaT] => [Tamashiro]

```

## 考察

文字反転を行うことは、文字配列の最後の文字から出力されることが分かる。  
 大文字・小文字変換プログラムとサンプルプログラムを応用する事で、作ることができた。  
 get\_n 関数で文字数を調べます。その文字数文 while 文で\*argvを進めると、最後の文字にたどり着き、そこから\*destに出力して行きます。

[ABC]という文字を与えたとします



## 参考サイト

初心者のためのポイント学習 C 言語

<http://www9.plala.or.jp/sgwr-t/index.html>

e0857ポチギ13

<http://www.ie.u->

[ryukyu.ac.jp/~e085713/home/report/syorui/c\\_rep6.pdf](http://ryukyu.ac.jp/~e085713/home/report/syorui/c_rep6.pdf)