

# プログラミング I

Report#5

提出日:2006年某日(木)

所属 :工学部情報工学科

学籍番号: 065702G

氏名 : 新垣智規

◎ 問題. ライブラリ関数 `islower()`, `toupper()`を使い、下記の `trlowup` プログラムを書き換えて、新規に `trupper` プログラムを作成せよ。

ソースコード

```
/*                                     Program : trupper.c
Comments : translate lower case characters into upper case ones.          */

#include <stdio.h>

char trupper(char);

int main(){
    char c;

    while( (c=getchar()) != EOF )
        putchar( trupper(c) );
    return(0);
}

char trupper( char c ){
    if (islower(c) == 0 )
        return(c);
    else
        return(toupper(c));
}
```

◎実行結果

```
[nw0602:~] j06002% ./trupper
AraKaKI
ARAKAKI
```

```
Onigiri oMUSubI
```

```
ONIGIRI OMUSUBI
```

```
[nw0602:~] j06002% ./trupper < data1
```

```
ABCDEFGHIJKLMNopQRSTUVWXYZ
```

## ◎考察

このプログラムは、入力された文字を判別し、小文字を大文字、大文字はそのまま表示するというプログラムである。

if 文を使い、まず大文字であるかないかを判別する。

その際、`islower()` を用いて判別をした。

if 文の中にある関数 `islower` は、大文字だと=0、それ以外は 0 以外の値を取る関数なので、それを利用し、`islower(c)=0` の時、すなわち変数 `c` の値が大文字の時は `retrn(c)`、`c` の値をそのまま返す（表示する）という書き方をした。

また、次に `else`、つまり `c` が小文字の場合は「`toupper`」関数で大文字に変換して返す（表示する）ので、入力した値はすべて大文字に直されて出力される。実行結果も予想通りの結果で、小文字を入力しても大文字を入力しても、どちらでも大文字出力した。

問題. `trupper` プログラムを書き換えて、`rot13` 暗号化プログラム、`rot13` 復号化プログラムを作成せよ。`rot13` とは、次のようにアルファベットを 13 文字ずらす暗号化の方法である。（暗号化と復元が一緒になる

ソースコード

```
/*                                                    Program : rot13.c
*/

#include <stdio.h>

char rot13(char);

int main(){
    char c;
```

```

while( (c=getchar()) != EOF )
    putchar( rot13(c) );
return(0);
}

char rot13( char c ){
    if (('a' <= c && c <= 'm') || ('A' <= c && c <= 'M'))
        return(c+13);
    else if (('n' <= c && c <= 'z') || ('N' <= c && c <= 'Z'))
        return(c-13);
    else
        return(c);
}

```

## ◎実行結果

```
[tomonori-arakaki-no-ibook-g4:~] j06002% ./rot13 < data1
```

```
nopqRSTuvwXyZAbCdEfghIjKlM
```

```
[tomonori-arakaki-no-ibook-g4:~] j06002% ./rot13
```

```
abcdEFGhijklMNopQRstUvWxyZ
```

```
nopqRSTuvwXYZAbcDEfgHiJklM
```

```
nopqRSTuvwXYZAbcDEfgHiJklM
```

```
abcdEFGhijklMNopQRstUvWxyZ
```

Program

Cebtenz

Cebtenz

Program

◎ 考察

rot13 プログラム、簡単にいうとアルファベットの前半部分と後半部分を入れ替えるプログラムで、a-m の値を n-z の値へそれぞれ変換、逆に n-z の値は a-m の値へ変換するという内容である。

大文字も同様に機能するように、それぞれ「||」(論理和)を使い、端的にまとめてみた。

これではじめは4行のif文が、2行に省略ができた。

このプログラム自体はとても簡単な作りになっており、アルファベット26文字を 'a' - 'm'、'n' - 'z' の2つに分け、前者の値の c が入力された場合、return(c+13)、つまり c のアスキーコードに割り振られている数に+13し、後者の場合は return(c-13)、c の値を-13するというプログラムということが書かれている。

- ・ ◎問題. オリジナルの暗号化・復号化プログラムを作成せよ。

ソースコード

```
/*                                                    Program : Angou.c
*/

#include <stdio.h>

char Angou(char);

int main(){
    char c;

    while( (c=getchar() ) != EOF)
        putchar(Angou(c));

    return(0);
}
```

```

char Angou(char c){
    if( 33<= c && c <=56)
        return(c+70);
    else if(57<= c && c <=79)
        return(c+23);
    else if(80<= c && c <=102)
        return(c-23);
    else if(103<= c && c <=126)
        return(c-70);
    else
        return(c);
}

```

## ◎ 実行結果

I Like Program and Computer !

` c#%N 9,)!,J' J(M Z)'\*/.N, g

` c#%N 9,)!,J' J(M Z)'\*/.N, g

I Like Program and Computer !

We have Mac OS X !

@N "JØN dJL f< A g

@N "JØN dJL f< A g

We have Mac OS X !

My name is Tomonori Arakaki.

d3 (J'N #- =)' )C),# X,J%J%#t

d3 (J'N #- =)' )C),# X,J%J%#t

My name is Tomonori Arakaki

## ◎ 考察

オリジナルの暗号&復号化プログラム。

ASCII コード表を参考にしながら、表示できる文字を入れ替えてみた。

まず、表示可能コードを 4 つのパートに分け、最初のパートと最後のパート、中間の 2 つのパートをそれぞれ入れ替える暗号復号化プログラムである。

つまり、表示可能な文字番号は 33-126 までなので、33-56 の文字のパートを 103-126 の文字パートに変換し、また逆に 103-126 も 33-56 に変換する。同じ要領で 57-79 と 80-102 を変換するといったプログラムになっている。

技術的にはあ rot13 プログラムを参考に、改良しただけだが、この暗号化は表示可能文字すべてに対応している。

## ◎ 反省・泣き言

中間時期と被ってしまい、忙しい 1 週間でした。

プログラムの課題をやる時間が、勉強の時間でどうしても減ってしまい、思った通りにオリジナル暗号プログラムができなかったことが一番心残りです。

最初の課題を先生が授業時間で終わる、的なことを言っていましたが、考えても授業時間では終わらず、rot13 が終わったのが週明けの月曜日といった悲惨さでした。

その上、試験勉強も加わってダウン寸前・・・大学始まって一番忙しかったです。

課題として、オリジナルのプログラムを仕上げましたが、まだまだ考え足りないなど反省しています。ただ+や-だけでなく、乗算や余を使った暗号化プログラムを制作したかった。。

先生がいつか言っていた時間は作るもの、との言葉が実感を伴って感じられました。少なくとも、ちゃんと予習復習していたらもう少しは時間をかけて作れたなあと。

この課題の反省点は、時間の使い方。これからうまく使えるように努力したいです。