

1. 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

```
1. #include <stdio.h>
2. #include <ctype.h>
3.
4. char trupper(char);
5. int main(){
6.     char c;
7.     while(( c = getchar()) != EOF){
8.         putchar(trupper(c));
9.     }
10.    return(0);
11. }
12.
13. char trupper(char c)
14. {
15.     if(islower(c) != 0)
16.         return(toupper(c));
17.     else
18.         return(c);
19. }
```

結果

```
Abcdefg    AbCdEfG    ABCDEFG
ABCDEFGG   ABCDEFGG   ABCDEFGG
123456789
123456789
!#$%&'()=~|
!#$%&'()=~|
```

考察

- 2 行目 `<ctype.h>` は 15 行目にある `islower` を使うために必要なヘッダー。
- 4 行目は `char` 型の `trupper()` 関数の設定。 `Char` 型の変数は文字を扱う時に使う。
- 6 行目、 `char` 型の変数 `c` の設定

7 行目:while 文による、繰り返し処理の実行。c が文字を取得、EOF でなければ繰り返す。
EOF とは End Of File の略であり-1 を返す。Visual Studio 2008 コマンド プロンプト環境
では、「control + C」で終了した。「control + Z」でも終わる。

8 行目は 4 行目で設定した trupper()関数を呼び出し、値の出力をする。

13 行目:trupper 関数の設定

15 行目:islower():アルファベット小文字の判定する。入力した文字が真なら 0 以外、偽ならば 0
を返す。この部分がサンプルプログラムの if('a' <= c && C <= 'z')に当たると考えられる。

16 行目:アルファベットの小文字を大文字に変換して値を返す。ここは「c - 'a' + 'A」に当たる。
ASCII コード表で'a'-'A'は 16 進数で 20、10 進数で 32 なのでここを「c - 32」や「c - 0x20」
と変えても同じ結果が出ると考えられる。

18 行目:上の条件以外なら c の値をそのまま返す。

Islower(c)は上で述べたようにアルファベット小文字の判定する関数。似たものにアルファベッ
ト大文字の判定する関数 isupper(c)がある。これも<ctype.h>ヘッダーが必要である。

toupper(c)は c が小文字なら大文字に変換した値を出す。具体的な働きは上記で述べたような働
きと考えられる。似たものに tolower がありこれは大文字の値を小文字の値に変換する。

また return(c + 32)、return(c + 0x20)でも同じ働きをすると予想される。

おまけ 1

上記のプログラムにおいて control + C 以外にどんな反応を示すか調べてみた。

ただし「^A」は「control + a」を意味する。

終了

^C ^Z(^C と違い Enter を押す必要があった)

文字が出るもの

^A ^B ^D ^E ^F ^G ^K ^L ^N ^O

^P ^Q ^R ^T ^U ^V ^W ^X ^Y

文字化けするので結果は載せませんでした

反応なし

^I ^S

Backspace キーみたいな働きをするもの

^H

Tab キーみたいな働きをするもの

^J

Enter キーみたいな働きをするもの

^M

おまけ 2

2 行付け足し大文字を小文字に変えるようにしてみた。

```
1. #include <stdio.h>
2. #include <ctype.h>
3.
4. char trupper(char);
5. int main(){
6.     char c;
7.     while(( c = getchar()) != EOF){
8.         putchar(trupper(c));
9.     }
10. return(0);
11. }
12.
13. char trupper(char c)
14. {
15.     if(islower(c) != 0)
16.         return(toupper(c));
17.     else if(isupper(c) != 0)  /*付け足した/*
18.         return(tolower(c)); /*付け足した/*
19.     else
20.         return(c);
21. }
```

結果

```
abcdefg ABCDEFG AbCdEfG
ABCDEFGG abcdefg aBcDeFg
1234567890
1234567890
!"#$%&'()=~|-^¥`{@[+*};:]<>?_./
!"#$%&'()=~|-^¥`{@[+*};:]<>?_./
```

考察

else if を付け足し isupper(c) で入力された文字が大文字と判別できるようにした。

また 18 行目で大文字の値を小文字の値に変えるようにした。

2. `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。

```
A --> N    a --> n
B --> O    b --> o
C --> P    c --> p
      :      :
Z --> M    z --> m
```

ソース

```
1. #include <stdio.h>
2. #include <ctype.h>
3.
4. char trupper(char);
5. int main(){
6.     char c;
7.     while(( c = getchar()) != EOF){
8.         putchar(trupper(c));
9.     }
10.    return(0);
11. }
12.
13. char trupper(char c)
14. {
15.     if(islower(c) != 0)
16.         if('a' <= c && c <= 'm')
17.             return(c + 0x0d);
18.         else
19.             return(c - ('n'-'a'));
20.     else if(isupper(c) != 0)
21.         if('N' <= c && c <= 'Z')
22.             return(c - 13);
23.         else
24.             return(c + ('/'-''));
25.     else
26.         return(c);
27. }
```

結果

```
abcdefghijklmnopqrstuvwxy  
nopqrstuvwxyzabcdefghijklm  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
NOPQRSTUVWXYZABCDEFGHIJKLM  
1234567890  
1234567890  
!#$%&'()=~|`{+*}>?_¥/.;:@^~ ( ° - ° )  
!#$%&'()=~|`{+*}>?_¥/.;:@^~ ( ° - ° )
```

考察

1~14行目は1と同じ

15行目で `islower` により入力したものがアルファベット小文字だった場合 16行目へ進むようになっている

16、17行目は15行目で真だった物、つまりアルファベット小文字に対して区分けをしている変数 `c` が `a` から `m` にあるとき `c` に `0x0d` つまり 13 を足して返す。

18、19行目それ以外つまり、変数 `c` が `n`~`z` にあるとき ASCII コードの `'n'` から `'a'` を引いた値つまり 13 を引いて返す。

16行目の `if` と 18行目の `else` がセットである。

20行目:アルファベット小文字以外で入力されたものに対し、`isupper` によりアルファベット大文字の判別し、アルファベット大文字だった物は 21行目に進むようになっている。

21、22行目:アルファベット大文字のうち `N`~`Z` の `c` に対し 13 が引かれ値を返す。

23、24行目:それ以外つまりアルファベット大文字のうち `A`~`M` に対し ASCII コードの `'/` から `“` を引いた値つまり 13 が引かれ値を返す。

21行目の `if` と 23行目の `else` はセットである。

25行目でアルファベット小文字とアルファベット大文字以外の文字に対し、値をそのまま返すようになっている。

15行目の `if` ,20行目の `else if` ,25行目の `else` がセットである。

3. オリジナルの暗号化・復合化プログラムを作成せよ。

暗号化プログラム

暗号化名「circle52」

ソース

```
1.  #include <stdio.h>
2.  #include <ctype.h>
3.
4.  char trupper(char);
5.  int main(){
6.      char c;
7.      while(( c = getchar()) != EOF){
8.          putchar(trupper(c));
9.      }
10.     return(0);
11. }
12.
13. char trupper(char c)
14. {
15.     if(isupper(c) != 0)
16.         if(c=='A')
17.             return('g');
18.         else if(c=='B')
19.             return('f');
20.         else if(c=='C')
21.             return('e');
22.         else if(c=='D')
23.             return('i');
24.         else if(c=='E')
25.             return('k');
26.         else if(c=='F')
27.             return('h');
28.         else if(c=='G')
29.             return('j');
30.         else if(c=='H')
31.             return('o');
32.         else if(c=='I')
33.             return('n');
34.         else if(c=='J')
35.             return('m');
```

```
36.         else if(c=='K')
37.             return('l');
38.         else if(c=='L')
39.             return('p');
40.         else if(c=='M')
41.             return('s');
42.         else if(c=='N')
43.             return('r');
44.         else if(c=='O')
45.             return('q');
46.         else if(c=='P')
47.             return('u');
48.         else if(c=='Q')
49.             return('v');
50.         else if(c=='R')
51.             return('t');
52.         else if(c=='S')
53.             return('w');
54.         else if(c=='T')
55.             return('c');
56.         else if(c=='U')
57.             return('z');
58.         else if(c=='V')
59.             return('x');
60.         else if(c=='W')
61.             return('y');
62.         else if(c=='X')
63.             return('a');
64.         else if(c=='Y')
65.             return('d');
66.         else
67.             return('b');
68.     else if(islower(c) != 0)
69.         if(c=='a')
70.             return('F');
71.         else if(c=='b')
```

```
72.             return('D');
73.     else if(c=='c')
74.             return('G');
75.     else if(c=='d')
76.             return('A');
77.     else if(c=='e')
78.             return('K');
79.     else if(c=='f')
80.             return('H');
81.     else if(c=='g')
82.             return('I');
83.     else if(c=='h')
84.             return('M');
85.     else if(c=='i')
86.             return('O');
87.     else if(c=='j')
88.             return('L');
89.     else if(c=='k')
90.             return('N');
91.     else if(c=='l')
92.             return('Q');
93.     else if(c=='m')
94.             return('R');
95.     else if(c=='n')
96.             return('P');
97.     else if(c=='o')
98.             return('S');
99.     else if(c=='p')
100.            return('U');
101.    else if(c=='q')
102.            return('W');
103.    else if(c=='r')
104.            return('V');
105.    else if(c=='s')
106.            return('T');
107.    else if(c=='t')
```



```

108.             return('Y');
109.         else if(c=='u')
110.             return('Z');
111.         else if(c=='v')
112.             return('X');
113.         else if(c=='w')
114.             return('E');
115.         else if(c=='x')
116.             return('J');
117.         else if(c=='y')
118.             return('C');
119.         else
120.             return('B');
121.     else
122.         return(c);
123. }

```

結果

AgInPuZbDiOqWyCeRlQvXaFhMsTcGjLpUzBfHoSwEkNrVxJmRtYd
gInPuZbDiOqWyCeKtQvXaFhMsTcGjLpUzBfHoSwEkNrVxJmRtYdA

考察

A→g→I→n→P→u→Z→b→D→i→O→q→W→y→C→e→R→l→Q→v→X→a→F→h→
M→s→T→c→G→j→L→p→U→z→B→f→H→o→S→w→E→k→N→r→V→x→J→m→R→t→
Y→d→A

上記のようになるように if 文 else if, else, を使いソースを組んだ。

15～67 行目

15 行目によりアルファベット大文字のみ 16 行目に進む。

16～65 行目

アルファベット大文字の A～Y までを別の文字に変換させる。

66、67 行目

16～65 行目までの書かれていないアルファベット大文字つまり Z のみ、b に変換する。

68～120 行目

68 行目によりアルファベット小文字を選別。

69～118 行目

アルファベット小文字 a から y を別の文字に変換させる。

119～120 行目

それ以外のアルファベット小文字つまり z を B に変換させる。

121～122 行目

アルファベット小文字、アルファベット大文字以外の文字は、そのまま値を返す。

大文字は小文字に、小文字は大文字になるようにソースも組んだ。

52 回ある文字を暗号化し続けると元に戻るようにした。

ので、復合化プログラムを作る必要はないが 52 回やるのは、大変なので一様ソースを下記に記す。

復号化プログラム

上記における 15 行目以下を下記に置き換える

```
1.  char trupper(char c)
2.  {
3.      if(isupper(c) != 0)
4.          if(c=='A')
5.              return('d');
6.          else if(c=='B')
7.              return('z');
8.          else if(c=='C')
9.              return('y');
10.         else if(c=='D')
11.            return('b');
12.         else if(c=='E')
13.            return('w');
14.         else if(c=='F')
15.            return('a');
16.         else if(c=='G')
17.            return('c');
18.         else if(c=='H')
19.            return('f');
20.         else if(c=='I')
21.            return('g');
22.         else if(c=='J')
23.            return('x');
24.         else if(c=='K')
```

```
25.             return('l');
26.     else if(c=='L')
27.             return('j');
28.     else if(c=='M')
29.             return('h');
30.     else if(c=='N')
31.             return('k');
32.     else if(c=='O')
33.             return('i');
34.     else if(c=='P')
35.             return('n');
36.     else if(c=='Q')
37.             return('l');
38.     else if(c=='R')
39.             return('m');
40.     else if(c=='S')
41.             return('o');
42.     else if(c=='T')
43.             return('s');
44.     else if(c=='U')
45.             return('p');
46.     else if(c=='V')
47.             return('r');
48.     else if(c=='W')
49.             return('q');
50.     else if(c=='X')
51.             return('v');
52.     else if(c=='Y')
53.             return('t');
54.     else
55.             return('u');
56. else if(islower(c) != 0)
57.     if(c=='a')
58.             return('X');
59.     else if(c=='b')
60.             return('Z');
```

```
61.         else if(c=='c')
62.             return('T');
63.         else if(c=='d')
64.             return('Y');
65.         else if(c=='e')
66.             return('C');
67.         else if(c=='f')
68.             return('B');
69.         else if(c=='g')
70.             return('A');
71.         else if(c=='h')
72.             return('F');
73.         else if(c=='i')
74.             return('D');
75.         else if(c=='j')
76.             return('G');
77.         else if(c=='k')
78.             return('E');
79.         else if(c=='l')
80.             return('R');
81.         else if(c=='m')
82.             return('J');
83.         else if(c=='n')
84.             return('I');
85.         else if(c=='o')
86.             return('H');
87.         else if(c=='p')
88.             return('L');
89.         else if(c=='q')
90.             return('O');
91.         else if(c=='r')
92.             return('N');
93.         else if(c=='s')
94.             return('M');
95.         else if(c=='t')
96.             return('R');
```

```

97.         else if(c=='u')
98.             return('P');
99.         else if(c=='v')
100.            return('Q');
101.        else if(c=='w')
102.            return('S');
103.        else if(c=='x')
104.            return('V');
105.        else if(c=='y')
106.            return('W');
107.        else
108.            return('U');
109.    else
110.        return(c);
111. }

```

非常に長くはなるが、やっていることは単純明快である。
上記のだけでは物足りないのもう一つ作りました。

プログラム 2

```

1.  #include <stdio.h>
2.  #include <ctype.h>
3.
4.  int mode;
5.  char upper(char);
6.  char downner(char);
7.
8.  int main(){
9.      char c;
10.     printf("何をしますか？\n");
11.     printf("1.暗号化\n");
12.     printf("2.復号化\n");
13.     printf("3.終了\n");
14.     scanf("%d",&mode);
15.     if(mode==1){

```

```

16.             printf("暗号化させる英文を入力してください。¥n[終了させるときは「control+c」を
                押してください]¥n");
17.             while((c=getchar()) !=EOF){
18.                 putchar(upper(c));
19.             }
20.         }
21.     else if(mode==2){
22.         printf("復号化させる英文を入力してください。¥n[終了させるときは「control+c」を
                押してください]¥n");
23.         while((c=getchar()) !=EOF){
24.             putchar(downner(c));
25.         }
26.     }
27.     else if(mode==3){
28.         printf("終了します¥n");
29.         return(0);
30.     }
31.     else
32.         printf("error¥n");
33.         return(0);
34.     return(0);
35. }
36. char upper(char c){
37.     if(isupper(c)){
38.         c+=10;
39.         if(c>'Z'){
40.             c-=26;
41.         }
42.         return(c);
43.     }
44.     else if(islower(c)){
45.         c-=10;
46.         if(c<'a'){
47.             c+=26;
48.         }
49.         return(c);

```

```

50.     }
51.     else
52.         return(c);
53. }
54. char downner(char c){
55.     if(isupper(c)){
56.         c-=10;
57.         if(c<'A'){
58.             c+=26;
59.         }
60.         return(c);
61.     }
62.     else if(islower(c)){
63.         c+=10;
64.         if(c>'z'){
65.             c-=26;
66.         }
67.         return(c);
68.     }
69.     else
70.         return(c);
71. }

```

結果

(1 を選択)

何をしますか？

1.暗号化

2.復号化

3.終了

1

暗号化させる英文を入力してください。

[終了させるときは「control+c」を押してください]

ABCDE QRSTU abcde qrstu 0123

KLMNO ABCDE qrstu ghijk 0123

(2 を選択)

C:\Program Files\Microsoft Visual Studio 9.0\VC\report4>angouka

何をしますか？

- 1.暗号化
- 2.復号化
- 3.終了

2

復号化させる英文を入力してください。

[終了させるときは「control+c」を押してください]

KLMNO ABCDE qrstu ghijk 0123

ABCDE QRSTU abcde□ qrstu 0123

(3 を選択)

C:\Program Files\Microsoft Visual Studio 9.0\VC\report4>angouka

何をしますか？

- 1.暗号化
- 2.復号化
- 3.終了

3

終了します

(その他の文字を入力)

C:\Program Files\Microsoft Visual Studio 9.0\VC\report4>angouka

何をしますか？

- 1.暗号化
- 2.復号化
- 3.終了

asdfg

error

考察

4 行目:変数 mode 設置

5、6 行目 char 型の関数 upper と downner の設定。

9 行目:変数 c の設定。

10～13 行目 printf により””の中を表示

14 行目:scanf により mode に整数値を取り込む。

先ほど入力した値により分岐

15 行目～20 行目

15 行目:mode の値が 1 の場合。

16 行目 printf により””の中を表示

17、18 行目 c が EOF でなければ関数 upper(c)に代入。

21 行目～26 行目

21 行目:mode の値が 2 の場合。

22 行目 printf により””の中を表示

23、24 行目 17、18 行目と同じただし代入先は関数 downner(c)である。

27 行目 mode の値が 3 の場合

28～30 行目：終了と表示させ終了させる

31～33 行目：error と表示させ終了。

36～53 行目：関数 upper のサブルーチン(暗号化させる)

値が入った場合この中の法則に従って c の値を変化、出力させる。

37～43 行目

c の値がアルファベット大文字の場合に対する c の値の変化

c の値から 10 を足す。もし c の値が'Z'よりも大きくなったら 26 を引いて

c の値になる。(c の値を'A'～'Z'にするためのもの)

44～50 行目

c の値がアルファベット小文字の場合に対する c の値の変化

c の値から 10 を引く。もし c の値が'a'よりも小さくなったら 26 で足して

c の値になる。(c の値を'a'～'z'にするためのもの)

50～53 行目

その他の文字はそのまま値を返す。

54～71 行目：関数 downner のサブルーチン(復号化させる)

値が入った場合この中の法則に従って c の値を変化、出力させる(やることは upper の反対)。

55～61 行目

c の値がアルファベット大文字の場合に対する c の値の変化

c の値から 10 を引く。もし c の値が'A'よりも小さくなったら 26 を足して

c の値になる。(c の値を'A'～'Z'にするためのもの)

62～68 行目

c の値がアルファベット小文字の場合に対する c の値の変化

c の値から 10 を足す。もし c の値が'Z'よりも大きくなったら 26 を引いて

c の値になる。(c の値を'a'～'z'にするためのもの)

69～71 行目

その他の文字はそのまま値を返す。

Error 報告

ついに暴走させました。Mode のところで else の設置を忘れてしまい、止まらなくなりました。最近「;」の付け忘れが少なくなりましたが、if 文の時に付け忘れてたりして「;」がないと怒られます。

感想

今回は今までで一番、燃えました。でもテスト勉強もしなければならぬので、不完全燃焼です。数字も暗号化すればよかったとおもいましたが、時間がなくあきらめました。先輩のレポートをみると課題の 3 の自分で作成はすごくて自分と比べると自分のが、がっかりに見えます。もっと勉強をして、先輩みたいなプログラムを作れるように頑張りたいです。

参考資料

「c 実践プログラミング」

「新 The UNIX Super Text(上)」