

プログラミング I

Report#1

提出日 : 2008 年 5 月 8 日 (木)

所属 : 工学部情報工学科

学籍番号 : 085738G

氏名 : 知名 隆 弘

A:出力するメッセージを変更せよ

ソース :

```
/*
Program   : hello.c
Student-ID : 085738G
Author    : TAKAHIRO,China
UpDate    : 2008/04/25(Fri)
Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("It is NOT easy paper.¥n");

    return(0);
}
```

・これは本文を変更しただけであるから、特に問題なく実行できると思われる。

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% /bin/tcsh
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
It is NOT easy paper.
[chinatakahiro-no-macbook:~] e085738%
```

考察 :

- a. 問題なく本文を変更できた。
- b. 試しに NOT と easy の間に全角スペースをいれてみると、エラーが出る事無くコンパイルされ、実行すると It is NOT easy paper. と表示されちゃんと改行もされていた。
- c. b より、printf の命令文中に全角スペースが入っても、問題ないという事がわかった。

B:同じメッセージを3回、別々の行に出力せよ。その書

ソース :

```
/*
Program    : hello.c
Student-ID : 085738G
Author     : TAKAHIRO,China
UpDate    : 2008/04/25(Fri)
Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("It is NOT easy paper.¥nIt is NOT easy paper.¥nIt is NOT
easy paper.¥n");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
It is NOT easy paper.
It is NOT easy paper.
It is NOT easy paper.
[chinatakahiro-no-macbook:~] e085738%
```

考察 :

- a. エスケープシーケンスの改行を使い、1つの `printf` 命令で別々の行に文字を打ち出した。
- b. エスケープ文字の後の文字は特別扱いと言っていたので、`¥nIs` のように打つとエラーが出ると予想していたがエラーは出なかった。
- c. b よりエスケープ文字の後の有効文字はそれの1つあとだけだという事がわかった。

a. その式

ソース :

```
/*
  Program   : hello.c
  Student-ID : 085738G
  Author    : TAKAHIRO,China
  UpDate    : 2008/04/25(Fri)
  Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("It is NOT easy paper.¥fIt is NOT easy paper.¥fIt is NOT
easy paper.¥f¥
");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
It is NOT easy paper.
                It is NOT easy paper.
                        It is NOT easy paper.

[chinatakahiro-no-macbook:~] e085738%
```

考察 :

- a. エスケープシーケンスのフォームフィードを使った。趣旨とはちょっと違うかもしれないが、別々の行に表す事ができた。
- b. ¥f の意味は次のページの先頭らしいがよくわからない。どこから次のページなのか、

調べてみたが調べきれなかった。

c. ¥は一つだけでは表示されない事と¥”);ではダメで¥

“);ならば問題なくコンパイルできる事がわかった。¥の後は何か文字を入れないと無視されるらしい。

C: 「Hello,」と「C World!」を別々の行に出力せよ。

ソース :

```
/*
  Program    : hello.c
  Student-ID : 085738G
  Author     : TAKAHIRO,China
  UpDate    : 2008/04/25(Fri)
  Comment    : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("Hello,¥nC World!¥n");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
Hello,
C World!
[chinatakahiro-no-macbook:~] e085738%
```

考察 :

a. 別々の行に表すのでとりあえず¥n、改行のシーケンスを使ってみた。特に問題なく改行できると予想していたので、予想通りだった。

b. ¥r と¥b を使った事が無いのでちょっと使ってみた。

```

/*
  Program   : hello.c
  Student-ID : 085738G
  Author    : TAKAHIRO,China
  UpDate    : 2008/04/25(Fri)
  Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("Hello,¥rC World!¥n");
    printf("hello,¥bC World!¥n");
    printf("hello,¥r");

    return(0);
}

```

というソースで実行したところ、

```

[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
C World!
helloC World!
[chinatakahiro-no-macbook:~] e085738%

```

という結果が返ってきた。

予想では最初に

C World!Hello,

と返ってくると思ったが違ったので、他の二つをみて検証してみたところ、戻る系のエスケープシーケンスを使うと、戻る前に入力されていた文字は消えてその上に書かれると言う事がわかった。

C : その式

ソース :

```
/*
Program   : hello.c
Student-ID : 085738G
Author    : TAKAHIRO,China
UpDate    : 2008/04/25(Fri)
Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("Hello,¥fC World!¥n");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
Hello,
    C World!
[chinatakahiro-no-macbook:~] e085738%
```

考察 :

- a. **B** のその式と同じく、**¥f** を使った。一度できた事なので今回も問題なくできるだろうと言う予想し、成功した。
- b. **¥f** を使ったあと文字を左詰めにできないかというのが反省点。
- c. **¥r** または **¥b** を使って左詰めにするやり方を思いついたので、次に示す。

C : その参

ソース :

```
/*
  Program   : hello.c
  Student-ID : 085738G
  Author    : TAKAHIRO,China
  Update    : 2008/04/25(Fri)
  Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("Hello,¥f¥rC World!¥n");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
Hello,
C World!
[chinatakahiro-no-macbook:~] e085738%
```

考察 :

- a. 予想では、¥f を使うと少し文字がずれていくようなので¥r を使って一番始めに持つていけばいいと言う考えだった。結果は予想通りとなった。
- b. ¥b を使う方法はずれているマス分だけ¥b を使って左に寄せると言うものだ。

```
printf("Hello,¥f¥b¥b¥b¥b¥b¥b¥b¥bC World!¥n");
```

と言う風にやるとうまく左詰めできた。

D : printf("...") と printf("...\n") の違いについて延べよ
printf("...") のソース :

```
/*
Program   : hello.c
Student-ID : 085738G
Author    : TAKAHIRO,China
UpDate    : 2008/04/25(Fri)
Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("Hello,C World!");
    printf("Hello,C World!");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
Hello,C World!Hello,C World![chinatakahiro-no-macbook:~] e085738%
```

改行シーケンスが無いので、改行されずにそのまま文字を表示している。

最後の出力文字で改行しないと、[chinatakahiro-no-macbook:~] e085738%の部分がすぐ後に出てきてしまって非常に見にくい状態になるのでなるべく改行して見やすくしたい。

printf(“...¥n”)のソース :

```
/*
  Program   : hello.c
  Student-ID : 085738G
  Author    : TAKAHIRO,China
  UpDate    : 2008/04/25(Fri)
  Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("Hello,C World!¥n");
    printf("Hello,C World!¥n");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
Hello,C World!
Hello,C World!
[chinatakahiro-no-macbook:~] e085738%
```

ちゃんと改行されてそれぞれ違う行に文字が出力されている。
改行しないで出力した方よりも文字も揃っていて見やすくなった。

¥nの有無の比較

¥n無しの実行結果

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
Hello,C World!Hello,C World![chinatakahiro-no-macbook:~] e085738%
```

¥n有りの実行結果

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
Hello,C World!
Hello,C World!
[chinatakahiro-no-macbook:~] e085738%
```

考察：

a. 二つを比べると、¥nがある方はちゃんと改行されて出力されている。

b. printfを二つに分けて書いてみたが、今度は一つの関数中でやってみた。

```
printf("Hello,C World!¥nHello,C World!¥n");
```

と言う風書き直してやってみると

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
```

```
[chinatakahiro-no-macbook:~] e085738% ./a.out
```

```
Hello,C World!
```

```
Hello,C World!
```

```
[chinatakahiro-no-macbook:~] e085738%
```

の様に二つにわけても一つでも同じ結果が得られた。¥nの無い場合も同様にすると

```
printf("Hello,C World!Hello,C World!");
```

結果は

```
Hello,C World!Hello,C World![chinatakahiro-no-macbook:~] e085738%
```

同じになった。

E: 同じメッセージを 3 回、同一行に出力せよ。

ソース :

```
/*
Program   : hello.c
Student-ID : 085738G
Author    : TAKAHIRO,China
UpDate    : 2008/04/25(Fri)
Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("Hello,C World! Hello,C World! Hello,C World!");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
Hello,C          World!          Hello,C          World!          Hello,C
World! [chinatakahiro-no-macbook:~] e085738%
```

考察 :

a. 単純に同じ文章を 3 回続けて書いた。

b. 予想通り続けて出力された。

c. `¥t` を使っていないので、使ってみると

```
printf("Hello,C World!¥t Hello,C World!¥t Hello,C World!¥n");
```

結果は Hello,C World! Hello,C World! Hello,C World!

と出た。`¥t` を使うと、文字をそろえて出力する事ができる事がわかった。

F : 菱形模様(「*」を用いる)を出力せよ

ソース :

```
/*
Program   : hello.c
Student-ID : 085738G
Author    : TAKAHIRO,China
UpDate    : 2008/04/25(Fri)
Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("\t*\n\t\t***\n\t\t\t*****\n\t\t\t\t***\n\t\t\t\t\t*");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out

    *
    ***
    *****
    ***
    *

[chinatakahiro-no-macbook:~] e085738%
```

考察 :

- a. タブ、バックスペース、改行を使った。
- b. 一度タブで打ち出す場所をそろえてからバックスペースで菱形になるよう調節した。
- c. タブだけでやってみたが、左が揃っただけだったので、バックスペースを使った。

その式」

ソース :

```
/*
  Program    : hello.c
  Student-ID : 085738G
  Author     : TAKAHIRO,China
  UpDate    : 2008/04/25(Fri)
  Comment    : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf(" *¥n ***¥n*****¥n ***¥n *¥n");

    return(0);
}
```

実行結果 :

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
 *
 ***
 *****
 ***
 *
[chinatakahiro-no-macbook:~] e085738%
```

考察 :

- a. スペースを使って文字の出力場所を調節した。
- b. 文字を寄せただけなので問題ないと予想し、予想通りの結果が出た。
- c. スペースを使ってやると手間がかかるのでなるべくシーケンスを使いたい。

G:「*」を用いて、自分の好きな形を出力せよ。

ソース:

```
/*
Program   : hello.c
Student-ID : 085738G
Author    : TAKAHIRO,China
UpDate    : 2008/04/25(Fri)
Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("* *%f * *%b* *%t * *%n* *%n");

    return(0);
}
```

実行結果:

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
[chinatakahiro-no-macbook:~] e085738% ./a.out
* *
    * * * * *
* *
[chinatakahiro-no-macbook:~] e085738%
```

考察:

- 形を思いつけなかったので適当にエスケープシーケンスを使ってみた。
- 出力する前にどういう形になるか考えてみたが、予想してみた形とは少し違った。
- 二行目はもっと離れたところから出力されると思ったが、**%f** を使うと前の行の最後の文字すぐ下の行に移動してそこから出力するという事がわかった。

H: エラーについて考察せよ。

C の考察 b の操作を行ったとき、以下のソースをコンパイルしようとしたところエラーが出た。

```
/*
  Program   : hello.c
  Student-ID : 085738G
  Author    : TAKAHIRO,China
  UpDate    : 2008/04/25(Fri)
  Comment   : Used Easy Function printf()
*/

#include<stdio.h>

int main(){

    printf("Hello,¥rC World!¥n");
    printf("hello,¥bC World!¥n");
    printf("hello,¥r")

    return(0);
}
```

エラーメッセージは

```
[chinatakahiro-no-macbook:~] e085738% cc hello.c
hello.c: In function 'main':
hello.c:17: error: syntax error before 'return'
```

と言う風にでたのでソースの 17 行目の `return` の前を見てみたところ、`()` の後に;
無い事に気づいた。

エラーメッセージをよく見ればどこが間違っているのか細かく指定されている事がわかるのでしっかり読む事が大切だと思う

反省・感想

このレポートを作るにあたっての反省は、作成するのがかなり遅れたと言う事だ。最初に手を加えた日からかなりの時間が経ってから続きをやったのでコンパイルの仕方や出力の仕方を忘れていてちょっと手間取ってしまった。あとソースと実行結果で幅をとってしまって考察がちょっと薄くなってしまったのが残念（正直、何書けばいいかわからなかった笑）。最後に、あんまり Word を使わないのでレイアウトをいじる余裕が無かった事が心残りだ。

このレポートを作成して、正直大変だと思った。ソースを作るのはそんなにきつくはなかったが、考察で何を書けばいいかわからないという状況が何度もあった。でも、課題をこなしていくうちに必要な情報を調べて自分で実践することで知識を増やしつつ技術も向上できると思うので頑張りたい。

参考文献

『C 実践プログラミング第3版』 Steve Oualline 著 望月 康司 監修 谷口 功
訳 株式会社オライリージャパン (1998)発行