

# 1.printf()関数による標準出力

## i.例題 hello.c

---

```
/*  
Program   : hello.c  
Student-ID : 095736E  
Author    : Tamashiro,Kakeru  
UpDate    : 2009/04/30(Thu)  
Comment   : Used Easy Function printf()  
*/
```

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

```
int main(){  
  
    printf("\n");  
    printf("Hello, C World!\n");  
    printf("\n");  

```

---

## 例題の出力結果

---

```
nw0936:prog1 e095736$ ./a.out
```

```
Hello, C World!
```

```
nw0936:prog1 e095736$
```

---

ii.テキストPP.50【特殊な文字(エスケープシーケンス)】を適宜使い、例題を参考に次のよう出力せよ。

a.出力するメッセージを変更せよ。

printf(“この中の文章を変更”)する事により、出力メッセージを変更する。

```
printf("Hello, C World!\n");  
    ↓下線を変更  
printf("max factor\n");
```

e095736

## a. 出力結果

---

```
nw0936:prog1 e095736$ ./a.out
```

```
"max factor"
```

```
nw0936:prog1 e095736$
```

---

見やすい様に”(ダブルクォート)を使っています。

## b. 同じメッセージを3回、別々の行に出力せよ。

エスケープシーケンスの\n(改行)を使う事により、別々の行に出力する。

```
printf("\nmax factor\n\nmax factor\n\nmax factor\n\n");
```

文末(下線部分)に\nを使う

## b. 出力結果

---

```
nw0936:prog1 e095736$ ./a.out
```

```
"max factor"
```

```
"max factor"
```

```
"max factor"
```

```
nw0936:prog1 e095736$
```

---

この場合は、ソースコードは1行でやったが、3行使ってでも出来る。しかし、その場合でも、\nは必要である。使わなかった場合、

## b. 出力結果(失敗例)

---

```
nw0936:prog1 e095736$ ./a.out
```

```
max factor/max factor/max factor
```

```
nw0936:prog1 e095736$
```

---

上記の様な結果になってしまう。

## c. 「Hello,」と「C World!」を別々の行に出力せよ。

前述の問題bと同じ要領で\nを使って、別々の行に出力する。

```
printf("\nHello,\n\nC World!\n\n");
```

「Hello,」と「C World!」の間(下線部分)に\n使う。

e095736

### c.出力結果

---

```
nw0936:prog1 e095736$ ./a.out
```

```
"Hello,"  
"C World!"
```

```
nw0936:prog1 e095736$
```

---

d.printf("...")とprintf("...\n")の違いについて延べよ。  
まず、下記の様なソースコードを作ります。

```
printf("Hello,C World!\n\"01\"\n");  
printf("Hello,C World!\n\"01\"\n");
```

### d.出力結果

---

```
nw0936:prog1 e095736$ ./a.out
```

```
Hello,C World!  
"01"  
Hello,C World!"01"
```

```
nw0936:prog1 e095736$
```

---

Hello,C World!の後の01を使って説明します。  
01の前に\nを使うと、Hello,C World!の後が改行され、次の行に01が出力されました。  
次に\nを使わないで入力すると、Hello,C World!の後は改行されず、01は同じ行に出力されました。  
つまり、カーソルが改行されているか、いないかの違いである。

e.同じメッセージを3回、同一行に出力せよ。

printf("同じ文章を続けて3回入力")する事により、同一行に出力する。

```
printf("max factor/max factor/max factor\n");  
同じ文章(下線部分)を3回
```

### e.出力結果

---

```
nw0936:prog1 e095736$ ./a.out
```

```
max factor/max factor/max factor
```

e095736

```
nw0936:prog1 e095736$
```

<3>

---

別で、ソースコードを3行使ってやってみます。

```
printf("max factor");
printf("max factor");
printf("max factor");
```

上記の様に\nを使わないで入力します。

e.出力結果(別パターン)

---

```
nw0936:prog1 e095736$ ./a.out
```

```
max factor/max factor/max factor/
```

```
nw0936:prog1 e095736$
```

---

ソースコードは3行だが、\nを使っていないので、同一行に同じメッセージが3回出力された。

f.次のような菱形模様(「\*」を用いる)を出力せよ。

---

```
 *
***
*****
***
 *
```

---

printf(“\*とスペースを使って入力”)する事により、菱形模様を出力する。

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

5個の\*(アスタリスク)を基準に、3個は両端にスペースを1つずつ、1個は2つずつ入力する。

f.出力結果

---

```
nw0936:prog1 e095736$ ./a.out
```

```
 *
***
*****
***
 *
```

```
nw0936:prog1 e095736$
```

---

次に\t(タブ)と\b(バックスペース)を使って、菱形模様を出力する。

e095736

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

\tを使う事により8文字分先から、出力されるようになります。

\bを使って、\*が3個は1文字分バックさせ、5個は2文字分バックさせます。

f.出力結果(別パターン)

---

```
nw0936:prog1 e095736$ ./a.out
```

```
 *
***
*****
***
 *
```

```
nw0936:prog1 e095736$
```

---

g.「\*」を用いて、自分の好きな形を出力せよ。

英語の”V”を出力しようと思います。下記の様に入力して、スペースの部分で”V”を作ります。

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

7行ぐらいが丁度バランスよく見えます。

g.出力結果

---

```
nw0936:prog1 e095736$ ./a.out
```

```
* ***** *
** ***** **
*** ***** ***
**** ***** ****
***** ** *****
***** * *****
*****
```

```
nw0936:prog1 e095736$
```

---

1行ずつに分けて入力した方が作りやすい。

## h.エラーについて考察せよ。

\nのnを抜いてみてわざとエラーを起こし、どういうものが出力されるのかを試す。

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

1行目は間違いはなく、2行目(下線部分)に間違いを入力してみます。

## h.出力結果

---

```
6:prog1 e095736$ cc exa8.c  
exa8.c: In function 'main':  
exa8.c:14: error: missing terminating " character  
exa8.c:15: error: syntax error before 'return'  
6:prog1 e095736$
```

---

エラーが上記(下線部分)の様に出力されました。

1行目は、ソースプログラムの14行目の行末(ダブルクォートの1文字前)に文字が抜けているという意味です。

2行目は、ソースプログラムの15行目のreturnの直前で文法の間違いが起こっていることを意味しています。

このエラーの意味は、間違った入力箇所の位置をしめしています。

## 2.反省・感想

初めてのプログラミングでやりにくいことばかりではあったが、参考書などをきっちり把握しとけばできないことではないので、もっと慣れる様にプログラミングをどんどん使っていこうと思います。レポートの方も初めてなのでこのレポートを機に、しっかりと書ける様に練習していこうと考えています。

# プログラミング1

Report01

**提出日: 2009年5月7日(木)**

**所属: 工学部情報工学科**

**学籍番号: 095736E**

**氏名: 玉城 翔**