

計算機とアルゴリズムI 第4回

森 義之

岡山理科大学

printf 文

printf 文は標準出力に文字列や数値を出力する関数である。

書式

printf 文

printf 文は標準出力に文字列や数値を出力する関数である。

書式

```
printf(書式指定文字列, データの並び);
```

書式指定文字列とは、二重引用符で囲まれた一般文字列または変換指定をいう。

printf 文

printf 文は標準出力に文字列や数値を出力する関数である。

書式

printf(書式指定文字列, データの並び);

書式指定文字列とは、二重引用符で囲まれた一般文字列または変換指定をいう。

一般文字列とは、一般的に使用される文字列 (abc, あいうえお, 日本語 など) や 改行を意味する `¥n` などである。

printf 文

printf 文は標準出力に文字列や数値を出力する関数である。

書式

printf(書式指定文字列, データの並び);

書式指定文字列とは、二重引用符で囲まれた一般文字列または変換指定をいう。

一般文字列とは、一般的に使用される文字列 (abc, あいうえお, 日本語 など) や 改行を意味する `¥n` などである。

変換指定は、講義後半で説明をする。

printf 文

例 (printf 文の例.)

```
1 #include <stdio.h>
2 int main(void){
3     printf("Hello C world");
4     return 0;
5 }
```

printf 文

例 (printf 文の例.)

```
1 #include <stdio.h>
2 int main(void){
3     printf("Hello C world");
4     return 0;
5 }
```

実行結果

Hello C world

printf 文

例 (printf 文の例.)

```
1 #include <stdio.h>
2 int main(void){
3     printf("Hello C world");
4     return 0;
5 }
```

実行結果

Hello C world

printf 文の () 内にある ” ” で囲まれた書式指定文字列 (の中の一
般文字列) が表示されている。

課題 B.0.

以下のプログラムを入力し、実行結果のように出力されることを確認せよ (サクラエディタ、Visual Studio Code)。

課題 B.0.

```
#include <stdio.h>
int main(void){
    printf("こんにちは");
    printf("Cの世界！");
    return 0;
}
```

実行結果

こんにちはCの世界！

printf 文

2つの printf 文を書いても、改行されない。

改行を表示させるには、`¥n` を表示させればよい。

課題 B.1.

課題 B.0. のプログラムを修正して、実行結果が

実行結果

こんにちは
Cの世界！

となるようにせよ。

特殊な記号はそのままでは表示できない。例えば

`%, ¥, "`

の記号を表示させる場合、そのまま入力しても表示されない。

[特殊な記号を表示させる方法]

`% ... %%` または `¥%`

`" ... ¥"`

`¥ ... ¥¥`

なお、`¥`は`\`と表示されることがある。

課題 B.2.

実行結果が

実行結果

%は%%で表示する。

"は¥"で表示する。

¥は¥¥で表示する。

となるプログラムを作成せよ。

ただし、¥は\と表示されていてもよい。

水平タブと改行の例

```
printf("abc\tklm\nnop\ns\tt\tu\n");
```

```
printf("%tmno%n");
```

```
printf("123456¥t7890¥n");
```

```
printf("1234567¥t890¥n");
```

```
printf("12345678¥t90¥n");
```

実行結果

```

abc      klm
nop
s        t          u
          mno
123456   7890
1234567 890
12345678          90

```

printf 文

この printf 文の書式は

```
printf(書式指定文字列, データの並び);
```

であった。書式指定文字列とは、二重引用符で囲まれた
一般文字列または変換指定であった。それぞれ、

一般文字列 : abc , 123, 日本語, あいうえお, ¥t, ¥n など

変換指定 : %d や %s など [%+変換指定子] で構成

である。

変換指定

変換指定	意味
%o	整数定数を 8 進数で出力
%d	整数定数を 10 進数で出力
%x	整数定数を 16 進数で出力 (英小文字)
%X	整数定数を 16 進数で出力 (英大文字)
%ld	long 型数値を 10 進数で出力
%f	浮動小数点定数を標準形で出力
%e	浮動小数点定数を指数形式で出力 (英小文字)
%E	浮動小数点定数を指数形式で出力 (英大文字)
%c	文字として出力
%s	文字列として出力
%p	アドレスを出力

書式指定文字列

例 (数値を出力するプログラムの例.)

```
#include<stdio.h>
int main(void){
    printf("10 進数で表す%d は、8 進数で表すと%o で、
           16 進数で表すと%x です。¥n",30,30,30);
    return 0;
}
```

実行結果

10 進数で表す 30 は、8 進数で表すと 36 で、16 進数で表すと 1e
です。

書式指定文字列

書式指定文字列の中に、変換指定を複数指定した場合、データの並びの左から順に処理をしていく。

例 (数値のデータを出力するプログラムの例.)

```
1 #include <stdio.h>
2 int main(void){
3     int a, b; // int は整数型 (integer) 宣言を意味する
4     a=5;
5     b=3;
6     printf("%d と %d の和は %d\n", a, b, a+b);
7     return 0;
8 }
```

実行結果

5 と 3 の和は 8

文字コードの例

例 (文字のコード)

```
1 #include <stdio.h>
2 int main(void){
3     char c; //char は文字型 (character) 宣言を意味する
4     c='A';
5     printf("%c の文字コードは%d です。¥n",c,c);
6     return 0;
7 }
```

実行結果

A の文字コードは 65 です。

オプション

数の表示において、桁数や精度を指定するため以下のオプションが使用できる。

%フィールド幅指定子 . 精度指定子 変換指定子

フィールド幅指定子 で数値の出力幅を指定する。

精度指定子 で浮動小数点の小数以下の桁数を指定する。

桁数をそろえて表示する場合には有用である。

オプション

実際の使用例

- %d 標準用法 10 進数で出力する。
- %8d 8 桁の 10 進数で出力する 先行 0 は表示されない。
- %08d 8 桁の 10 進数で出力する 先行 0 が出力される。
- %-8d 左寄せで 8 桁の 10 進数を出力する。
- %f 標準用法 浮動小数を標準桁数で出力する。
- %10.2f 浮動小数を (小数点, 符号も 1 桁と数え) 全体 10 桁、
小数点以下 2 桁出力する。

課題 B.3.

以下のプログラムの(ア)~(ウ)に適当な文字を入れ、実行結果のように出力されるプログラムを作成せよ。

課題 B.3.

```
#include<stdio.h>
void main(void){
    int a=2;
    printf("a=%(ア)d,¥n",a);
    printf("a=%(イ)d,¥n",a);
    printf("a=%(ウ)d,¥n",a);
}
```

実行結果

```
a=  2,
a=2   ,
a=0002,
```

オプション

例 (フィールド幅指定子, 精度指定子を用いた表示例 2)

```
#include<stdio.h>
void main(void){
    double da=-12.3456789;
    printf("da=%f¥n",da);
    printf("da=%10.4f¥n",da);
    printf("da=%10.4f¥n",da*(-1));
}
```

オプション

実行結果

```
da=-12.345679
```

```
da=  -12.3457
```

```
da=   12.3457
```

3つ (da の値) はどれも (標準形式と同じ)10 桁で表示されている。
ただし、10 桁で表示できない場合はそれ以上の桁で表示される。

算術演算 (四則演算と余り)

例 (算術演算の例)

```
int a=10,b=3;  
printf("a+b=%d\n",a+b);  
printf("a-b=%d\n",a-b);  
printf("a*b=%d\n",a*b);  
printf("a/b=%d\n",a/b);  
printf("a%%b=%d\n",a%b);
```

実行結果

```
a+b=13  
a-b=7  
a*b=30  
a/b=3  
a%b=1
```

本日の課題

- ・ 課題 B.1.
- ・ 課題 B.2.
- ・ 課題 B.3.
- ・ 課題 B.4. (プリント 35 ページ 課題 2.4.1.)

提出方法はメールで `mori@xmath.ous.ac.jp` まで送る。

件名は学生番号と課題 B. と書き、

例) S24M999 課題 B.

ファイルを添付して送ること。本文は空白のままでよい。

※ファイル名注意！

締め切りは5月18日 23:59。