## C 言語検定必勝プリント 明日のために3級編

No.04 関数 printf の秘密

今回は今までとはうってかわって関数のお話になります。この「関数」ですが実はC言語の屋台骨大切な部分です。でもよくわからない部分でもあります。そこで、ここで関数は何か・・・と語ってしまってもいいのですがC検定3級の範囲を超えてしまいますので今回はその中でもすごくスタンダードな関数である printf (ぷりんとえふ)について語りましょう。

まずはこの printf をヘルプで引いてみると

int printf( const char \* format [, argument]...); 標準出力ストリームに書式付きで出力します。

???少しわかりやすく説明するとこんな感じになります。 「こちらが指定したとおりに画面に表示しますよ。」

指定したとおりってのが書式、画面ってのが標準ストリームと思ってください。ちなみに一番簡単な書式は多分文字列ということになるでしょう。 こんな感じ

printf("Hellow World!\feetan");

最後の¥n は制御コードといって文字の一種です'¥'は続く文字が実は普通の文字じゃなくて特殊な文字ですよということを表現するために使います。ちなみにここで使っている ¥n は「改行しなさい」ってことです。

でももう少し面白い指定ができます。それが変数の値に合わせて表示を変える指定です。ちなみにこんなことになります。

int a=100; printf ("aの値は %d どえす¥n",a);

#### 実行結果

a の値は 100 どえす

なにやら printf 関数の中の文字列に変な "%d" ってのが付きました。しかも実行結果を見ると%d が 100 に入れ替わっています。この%d ってのは「書式指定子」といい書式文字列の後に指定する変数の値に変えて表示してくれるものです。ちょっと表にまとめてみましょう。

書式指定子	データ型
%c	char
%s	文字列へのポインタ
%hd	short int
%d	int
%ld	long int
 %f	float
/01	double

これをうまく使えばこんなこともできます。

int a=100; printf ("aの値は %c どえす¥n",a);

### 実行結果

a の値は d どえす

これを関数のプロトタイプ宣言といいます。

※今はこの考え方はあまり面白くなさ そうですが、ネットワーク系のプログラ ミングなどするとありがたみが分かり ます。

¥n なんかをエスケープシーケンスって 言います。詳しい説明は教科書を読んで もらうとして代表的なものを書いてお きます。

エスケープ シーケンス	意味
¥n	改行
¥t	水平タブ
¥Ο	ヌル
¥v	垂直タブ
¥b	後退
¥r	復帰
¥f	改ページ
¥a	警告音

←文字コードはただの番号なので この場合コード番号 100 番の文字が 出力されます。

### C 言語検定必勝プリント確認問題 明日のために3級編

No.04 printf の秘密

氏名		
$\vdash$		

**1** 次の文章にある括弧に当てはまる語句を解答群より選び答えなさい。

printf は C の関数で、指定されたフォーマットで標準出力に文字を出力することができます。 VC6でヘルプを見ると printf は以下のように定義されています

int printf( const char \*format [, argument]...);

const char は「( a ) 文字」という意味合いになりますが直後に'\*'がありますので正確には「( a ) 文字列へのポインタ」となります。このあたりは次の配列で説明するとしてさっきの「( a )…」って所を考えて見ます。 普通に Hellow! と画面に出力する場合次のように書きますね

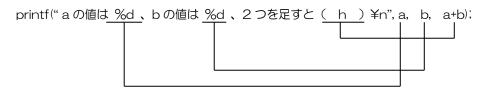
printf("Hellow!\fmathbf{y}n");

そうですね一度"Hellow!" と表示しなさいとソースに書くと後で変更するにはソースを書き換えるしかなくなりますね。でもそれだけでは困る場合があります。例えば計算結果を表示しようとするとき、同じ文字列しか表示できないと出力の数だけ printf を準備するか、毎回計算結果を手で計算しソースに埋め込まなくてはなりません。 非常に鬱陶しいです。マジで・・・

そこで文字列の中に置き換えができる部分を準備しておき、表示のたびにそこだけ入れ替えるということにしました。 その入れ替える部分の指定を行うものが( b )というものになります。大きく分けると文字、( c )、整数、( d ) と 4 つの表示に対応するように準備されていますが、数字などの表示を行う場合、桁合わせなどの必要性から細かい 指定ができるようになっています。まず基本の書式指定子を表にして見ました。

書式指定子	データ型
%(e)	char
%s	文字列へのポインタ
%hd	short int
% ( f )	int
%ld	long int
%(g)	float
/o ( g )	double

で実際にこれを使ってみるとこんな風になります。



これ以外にも桁あわせのための書式があったり、( i ) コード(制御コード)があったりします。そのおかげで数字が変わったりしてもいちいちプログラム変更のような面倒なことをしなくてもいいようになっています。

#### 【解答群】

ル)	а	
アノ		

イ) b

ウ)c ケ)エスプリ

I) d

コ) エスケープ

才)e

サ) 文字列

カ) f シ) 変数

キ)書式決定子ス)実際数

セ) 実数

ク) 書式指定子

ソ) 書き換えられる

タ) 書き換えれない

チ) %a

ツ) %b

テ) %c

ト) %d

а	Ь	С	d	е	
f	യ	h			

# 問2次の各設問の出力結果を記入しなさい。

1 2

3

## C 言語検定必勝プリント確認解答 明日のために3級編

No.04 printf の秘密

氏名		
$\nu_{-}$		

**PD 1** 次の文章にある括弧に当てはまる語句を解答群より選び答えなさい。

printf は C の関数で、指定されたフォーマットで標準出力に文字を出力することができます。 VC6でヘルプを見ると printf は以下のように定義されています

int printf( const char \*format [, argument]...);

const char は「( a ) 文字」という意味合いになりますが直後に'\*'がありますので正確には「( a ) 文字列へのポインタ」となります。このあたりは次の配列で説明するとしてさっきの「( a )…」って所を考えて見ます。 普通に Hellow! と画面に出力する場合次のように書きますね

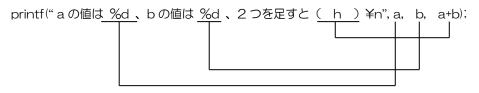
printf("Hellow!\fm");

そうですね一度"Hellow!" と表示しなさいとソースに書くと後で変更するにはソースを書き換えるしかなくなりますね。でもそれだけでは困る場合があります。例えば計算結果を表示しようとするとき、同じ文字列しか表示できないと出力の数だけ printf を準備するか、毎回計算結果を手で計算しソースに埋め込まなくてはなりません。 非常に鬱陶しいです。マジで・・・

そこで文字列の中に置き換えができる部分を準備しておき、表示のたびにそこだけ入れ替えるということにしました。その入れ替える部分の指定を行うものが( b )というものになります。大きく分けると文字、( c )、整数、( d ) と 4 つの表示に対応するように準備されていますが、数字などの表示を行う場合、桁合わせなどの必要性から細かい指定ができるようになっています。まず基本の書式指定子を表にして見ました。

書式指定子	データ型
%(e)	char
%s	文字列へのポインタ
%hd	short int
% ( f )	int
%ld	long int
%(g)	float
/o ( g )	double

で実際にこれを使ってみるとこんな風になります。



これ以外にも桁あわせのための書式があったり、( i ) コード(制御コード)があったりします。そのおかげで数字が変わったりしてもいちいちプログラム変更のような面倒なことをしなくてもいいようになっています。

#### 【解答群】

ア) a 1) b

ウ) c

c I

I) d

才) e

力) f

キ)書式決定子

ス)実際数

ク) 書式指定子 セ) 実数 ケ) エスプリ ソ) 書き換えられる

コ) エスケープ サ) 文字列 タ) 書き換えれない

シ)変数

チ) %a

ツ) %b

テ) %c

ト) %d

а	タ	b	ク	С	サ	d	セ	е	ウ
f	Н	യ	カ	h	7	i	П		

## 門2次の各設問の出力結果を記入しなさい。

1)
int a=10,b=5:
printf("a=%d b=%d a+b=%d",a,b,a+b):
3)
int a1=1,a2=10,a3=100:
printf("a1= %3d¥n",a1):
printf("a2= %3d¥n",a2):
printf("a3= %3d¥n",a3):

※教科書を調べてくださいね。

2)
char n1='A';
printf("A'+5=%c",n1+5):
4)
float f1=12.34;
printf("f1= %f¥n",f1):
printf("f1= %f¥n",f1):
printf("f1= %6.3f¥n",f1):

※教科書を調べてくださいね。

2)
char n1='A';
printf("A'+5=%c",n1+5):
4)
float f1=12.34;

1	а	=	1	0		В	=	5		а	+	b	=	1	5			
2		А	+	5	=	F												
	а	1	=			1												
3	а	2	=		1	0												
	а	3	=	1	0	0												
4	f	1	=	1	2	·	3	4	Ο	0	0	Ο						
4	f	1	=	1	2	·	3	4	0									