

プログラミング2

第十二回 標準ライブラリ関数の基礎

機械システム系 劉 子昂(LIU Ziang)

第6回レポート課題について

- 課題はMoodleを見て下さい. 課題1～課題2まであります.
- Moodleのページからソースファイル(.c)を提出(アップロード)してください.

標準ライブラリ関数の基礎

- 前処理指令 10章
 - 前処理指令 73
 - #include 74
 - #define 75
- 標準ライブラリ関数 11章
 - 文字列処理関数 79
 - 数学関数 81

前処理指令

使用例

```
#include <stdio.h>
#define SIZE 100
```

- コンパイル
 - 前処理 + コンパイル
- 前処理を行うのがプリプロセッサです。
- プリプロセッサ (preprocessor)
 - pre: 前もって、process: 処理する
- プリプロセッサは#で始まる行を処理する
- Cの文ではないので末尾に ; は付かない

#includeを使う

- `#include` は指定されたファイルをその場所に挿入する
- `#include <ファイル名>`
 - `<ファイル名>` は, ファイルを標準ディレクトリで探す
- `#include "ファイル名"`
 - `"ファイル名"` は, ファイルをまず作業ディレクトリから探す
 - 作業ディレクトリにない時には標準ディレクトリで探す
- 通常ヘッダファイル(.h)を読み込む
- `#include <stdio.h>`
 - `printf()`などの関数

使用例

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

#includeを使う

ファイル名 : test.c

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

int main(void){
    myfunc();
    printf("hdt=%d\n", hdt);
    return 0;
}
```

ファイル名 : myheadfile.h

```
int hdt = 1234;
void myfunc(void){
    printf("test program\n");
}
```

実行結果

```
test program
hdt=1234
```

#defineを使う

- オブジェクト形式マクロ
 - 書式: `#define` 識別子 置換要素並び
 - 単純な文字列の置き換え

```
#include <stdio.h>
#define PI 3.14159
#define FORMAT "%f\n"
int main(void){
    printf(FORMAT, 2*PI*3.0);
    return 0;
}
```

```
#include <stdio.h>

int main(void){
    printf("%f\n", 2*3.14159*3.0);
    return 0;
}
```

#defineを使う

- 関数形式マクロ

- 書式: `#define 識別子(引数) 置換要素並び`
- 引数を使って文字列の置き換えを行います

```
#include <stdio.h>
#define cube(x) ((x)*(x)*(x))
int main(void){
    printf("%d\n", cube(3));
    return 0;
}
```

```
#include <stdio.h>

int main(void){
    printf("%d\n", ((3)*(3)*(3)));
    return 0;
}
```


関数形式マクロの副作用

```
#define mul(a, b) a * b           //マクロ  
a = mul(x + 100, y + 200);      //記述  
a = x + 100 * y + 200           //置き換え結果
```

```
#define mul(a, b) (a) * (b)      //マクロ  
a = mul(x + 100, y + 200);      //記述  
a = (x + 100) * (y + 200)       //置き換え結果
```

```
#define add(a, b) (a) + (b)      //マクロ  
a = add(x, y) * 200;             //記述  
a = (x) + (y)*200                //置き換え結果
```

```
#define add(a, b) ((a) + (b))    //マクロ  
a = add(x, y) * 200;             //記述  
a = ((x) + (y)) * 200           //置き換え結果
```

文字列処理関数

```
#include <string.h>
```

関数	説明
<code>strcpy(s1, s2)</code>	文字列s2をs1にコピーする
<code>strncpy(s1, s2, n)</code>	文字列s2の先頭n文字をs1にコピーする
<code>strcat(s1, s2)</code>	文字列s1の後ろにs2を連結する
<code>strncat(s1, s2, n)</code>	文字列s1の後ろにs2の先頭n文字を連結する
<code>strcmp(s1, s2)</code>	文字列s1とs2を比較する。等しければ戻り値は0
<code>strncmp(s1, s2, n)</code>	文字列s1とs2の先頭n文字を比較する
<code>strlen(s1)</code>	文字列s1の長さを返す
<code>strchr(s1, ch)</code>	文字列s1内の文字chの位置(アドレス)を返す
<code>strchr(s1, s2)</code>	文字列s1内の文字列s2の位置(アドレス)を返す

数学関数

```
#include <math.h>
```

```
gcc test.c -o test.exe -lm -ansi -pedantic -Wall
```

- 数学関数における注意点
 - 関数の戻り値はdouble型
 - 関数の引数はdouble型
 - 角度の単位はラジアン

関数	意味
<code>sin(x)</code>	$\sin(x)$
<code>cos(x)</code>	$\cos(x)$
<code>tan(x)</code>	$\tan(x)$
<code>asin(x)</code>	$\sin^{-1}(x)$
<code>acos(x)</code>	$\cos^{-1}(x)$
<code>atan(x)</code>	$\tan^{-1}(x)$
<code>atan2(y, x)</code>	$\tan^{-1}(y/x)$
<code>fabs(x)</code>	$ x $

関数	意味
<code>exp(x)</code>	e^x
<code>log(x)</code>	$\ln(x)$
<code>log10(x)</code>	$\log_{10} x$
<code>pow(x, y)</code>	x^y
<code>sqrt(x)</code>	\sqrt{x}
<code>ceil(x)</code>	$\lceil x \rceil$
<code>floor(x)</code>	$\lfloor x \rfloor$

問題に挑戦！

1. 下線部を適切に埋めよ. ヒントは, 下記コメント文と p.200の表にある.

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char m[80], n[80];
    _____(m, "Hello! Mr.");    /* 文字配列mに文字列をコピーする */
    scanf("%s", n);
    _____(m, n);               /* mの後ろに文字列nを連結する */
    if(_____ (n, "Taro") != 0) {  /* nが"Taro"でないとき次の表示 */
        printf("%s¥n", m);
    }
    printf("Length=%d¥n", _____(m)); /*文字列mの長さを求めて表示*/
    return 0;
}
```

(さらにヒント: strcpy, strlen, strcat, strcmp のどれかが入る)

1. のプログラム例

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char m[80],n[80];
    strcpy(m,"Hello! Mr.");    /* 文字配列mに文字列をコピーする */
    scanf("%s",n);
    strcat(m,n);               /* mの後ろに文字列nを連結する */
    if(strcmp(n,"Taro")!=0){   /* nが"Taro"でないとき次の表示 */
        printf("%s\n",m);
    }
    printf("Length=%d\n", strlen(m)); /*文字列mの長さを求めて表示*/
    return 0;
}
```

Suzuki

Hello! Mr.Suzuki

Length=16

作ってみよう！

2. 角度 $0\sim 360^\circ$ (30° 刻み)についてsinの値を求めて出力せよ. (ヒント p.271)

3. 値 $0\sim 10$ (1刻み)について平方根の値を求めて出力せよ.

2.解答例

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    double a, d;
    for (a = 0.0; a <= 360.0; a += 30.0)
    {
        d = sin(a / 180.0 * 3.1416);
        printf("sin(%.0f deg)=%.5f\n", a, d);
    }
    return 0;
}
```

実行結果

```
sin(0 deg)=0.00000
sin(30 deg)=0.50000
sin(60 deg)=0.86603
sin(90 deg)=1.00000
sin(120 deg)=0.86602
sin(150 deg)=0.49999
sin(180 deg)=-0.00001
sin(210 deg)=-0.50001
sin(240 deg)=-0.86603
sin(270 deg)=-1.00000
sin(300 deg)=-0.86602
sin(330 deg)=-0.49999
sin(360 deg)=0.00001
```

3.解答例

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    double a, d;
    for (a = 0.0; a <= 10.0; a += 1.0)
    {
        d = sqrt(a);
        printf("root(%.0f)=%.8f\n", a, d);
    }
    return 0;
}
```

実行結果

```
root(0)=0.00000000
root(1)=1.00000000
root(2)=1.41421356
root(3)=1.73205081
root(4)=2.00000000
root(5)=2.23606798
root(6)=2.44948974
root(7)=2.64575131
root(8)=2.82842712
root(9)=3.00000000
root(10)=3.16227766
```