

1. hello.c 下記の横線の間の 6 行プログラムを vim よりこのファイル名で編集、コンパイル、実行

```
#include <stdio.h>
int main(void)
{
    printf("Hello world!¥n");
    return 0;
}
```

```
% gcc hello.c // コンパイル：gcc によりデフォルトでできる実行ファイルは a.out
% ./a.out // $PATH の中にカレントディレクトリはないので ./a.out とフルパスを指定
Hello world!
```

2. number.c 1 と同様

```
#include <stdio.h>
int main(void)
{
    int i;
    for(i=0;i<15;i++)
    {
        printf("%3d¥n", i+1);
    }
    return 0;
}
```

```
% gcc number.c -o num // オプションの後に実行形式の名前を指定すると a.out の代わりに num という
% ./num // 名前の実行形式ファイルができる
1
2
...
15
```

3. 課題 1

上記 2. の number.c では 1 から 15 までの整数値が出力されるが、これを編集して、3 の倍数の場合には数値のあとに ! を表示し、5 の倍数の場合には何も表示しないプログラムとし、ソースファイル名を name0601.c とし提出する。

注意 a. 山田太郎の学生番号の場合、Yamada0601.c (replace with your name) のようにしてファイル名に日本語を入れない。メールのタイトルは 山田太郎 0601 とする

注意 b. **提出するのは編集した .c のソースファイルで、コンパイル後の a.out (実行形式) ではない**

注意 c. ソースファイルをメールの添付書類とし、knakamura.maebit@gmail.com に送信

以下のように、number.c を提出様用ファイルにコピーして編集する

```
% cp number.c Yamada0601.c
% vi Yamada0601.c
```

4. mycat1.c 標準入力 of 文字列をそのまま標準出力に書き出す

```
#include <stdio.h>
int main(void)
{
    char c; // 1 文字ずつ読み込む時のキャラクタ
    while((c=getc(stdin))!=EOF) // while ループで読む文字がファイルの末尾になるまで
    { // 標準入力から一字ずつ読む
        putchar(c, stdout); // 読んだ文字を標準出力に書き出す
    }
    return 0;
}
```

```
% gcc mycat1.c -o mycat1 // 上記のソースコードがそのまま画面に出力される
% ./mycat1 < mycat1.c
```

5. mycat1.c を利用して、標準入力から読んだ文字列を出力せずに、標準入力に入った行数を表示する mylc1.c を作り提出しなさい

```
% gcc mylc1.c -o mylc1
% ./mylc1 < mycat1.c
10
```

6. 課題 2

5 で作成したプログラムを name0602.c と改名して knakamura.maebit@gmail.com に送信

山田太郎が作成したプログラムなら

```
cp mylc1.c yamada0602.c
```