

プログラミング 演習 I 報告書

題 目	1. 文字の出力
------------	----------

実施年月日 2024 年 10 月 01 日 (火 曜 日)

提出年月日 2024 年 10 月 07 日 (月 曜 日)

提出期限 2024 年 10 月 07 日 (月 曜 日)

実施場所 制御情報工学科 コンピュータ演習室

学 年	S2	番 号	13	氏 名	小野 涼大
-----	----	-----	----	-----	-------

(2) 問題分析	
(3) 設計	
(4) 実装	
(5) 検証	
(6) 考察	
合計	

1 問題設定

```
1 Hello.  
2 私は小野です。
```

という 2 行のメッセージを画面に表示する。

2 問題分析

今回の問題はターミナル上に英語と日本語のメッセージを表示させれば良い。まず「Hello.」と半角英語で表示して改行し、次に「私は小野です。」と全角日本語で表示して改行する。そしてプログラムを終了させれば良い。

本プログラム作成にあたり、標準入出力 (standard input output) ヘッダファイルの `<stdio.h>`, 文字列を表示する `printf`, 改行の制御文字 `\n` を用いる。

3 設計

以下に今回作成するプログラムのフローチャートを示す。

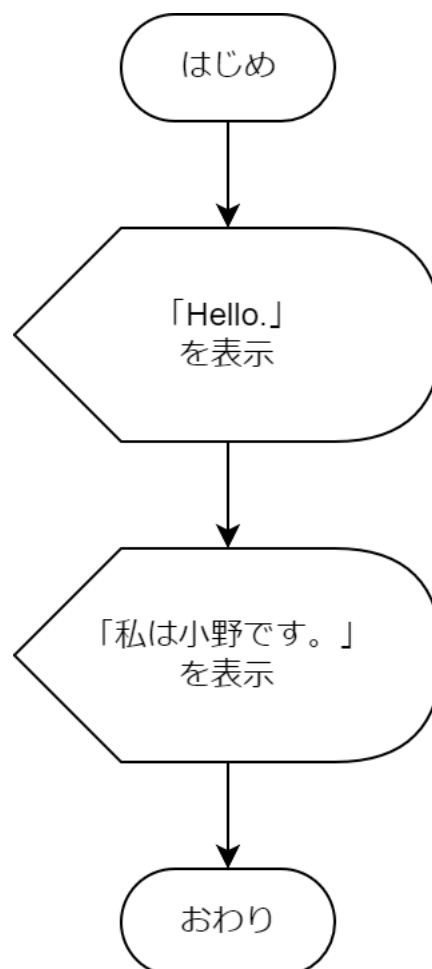


図1: フローチャート

4 実装

まず今回書いたコードを載せる.

1: ソースコード

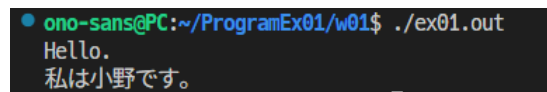
```
1  /******  
2  例題1 メッセージの表示 [ex01.c]  
3  *****/  
4  
5  #include <stdio.h> // standard input output ヘッダを読み込む  
6  
7  int main(void) // main関数の始まり  
8  {  
9      printf("Hello.\n");          // "Hello."を表示  
10     printf("私は小野です.\n");    // "私は小野です."を表示  
11     return 0;                    // 0を返し正常終了したことを伝える  
12 }
```

各行で行っていることについてはその都度コメントで記載してあるが、補足しておく必要があるものもあるので追加で説明する.

まず `#include <stdio.h>` で「standard input output ヘッダを読み込む」と書いてある. C 言語には関数についてまとめたファイルを読み込む機能があり, そのうちよく使われるもの (今回で言えば `printf`) などは標準ライブラリと呼ばれるものに入っており, そのうちのひとつとして標準入出力のファイルを読み込んでいる.

5 検証

実行結果は以下の通りである.



```
ono-sans@PC:~/ProgramEx01/w01$ ./ex01.out  
Hello.  
私は小野です.
```

図2: 実行結果

これは自分の Windows11 の PC に WSL2 を入れ, gcc でコンパイルしたものだ. 1 行目の「Hello.」はソースコード1の 9 行目, 「printf("Hello.\n");」, 1 行目の「私は小野です。」はソースコード1の 10 行目, 「printf(" 私は小野です.\n");」, にそれぞれ対応している.

6 考察

6.1 printf の結合

`printf` を 2 回に分けて文字を表示させているが, 繋げても表示できるのではないかと考え, 一つの `printf` で表した.

2: 1 つの printf で表したコード

```
1 #include <stdio.h>  
2  
3 int main(void)  
4 {  
5     printf("Hello.\n私は小野です.\n"); // 1つのprintfに結合  
6     return 0;  
7 }
```

表示された結果は頭2と同じような結果になった.

6.2 main 関数を text 関数に変更

`int main(void)` となっている部分を `main` 関数と呼ぶのであれば、`main` の部分を `test` に変更し `test` 関数 (?) にして実行してみた。

3: main を test に

```
1 #include <stdio.h>
2
3 int test(void) // main関数のmainの部分をtestに変更
4 {
5     printf("Hello.\n");
6     printf("私は小野です。 \n");
7     return 0;
8 }
```

コンパイルすると以下のように出力された。

```
ono-sans@PC:~/ProgramEx01/w01$ gcc -o ex01_02.out ex01.c
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/Scrt1.o: in function `_start':
(.text+0x1b): undefined reference to `main'
collect2: error: ld returned 1 exit status
```

図3: 実行結果

テキストに書き起こしたものを以下に載せる。

4: エラー文

```
1 ono-sans@PC:~/ProgramEx01/w01$ gcc -o ex01_02.out ex01.c
2 /usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/Scrt1.o: in function `_start':
3 (.text+0x1b): undefined reference to `main'
4 collect2: error: ld returned 1 exit status
```

エラーの内容を簡単に翻訳すると、自分の入力したところとそこから 1 行目、3 行目は自分が書いた内容との関連性はないと考えられる。2 行目を日本語に直すと「未定義の `main` への参照」となっていることから簡単に言えば「`main` がないよ」と返されていることがわかる。

このことから C 言語を実行するには `main` 関数が不可欠で、`main` 以外の名前では認識されないと考えられる。

[1] より

- `main` 関数は必須。
- `main` 関数から実行が開始される。

とのことから `main` 関数があればそれ以外の関数も書くことができると考えられる。

6.3 main 関数と text 関数

printf が上から実行されていったことから main 関数の下に test 関数というものを書いてみる。

5: main 関数と text 関数

```
1 #include <stdio.h>
2
3 int main(void) // main関数
4 {
5     printf("Hello.\n");
6     return 0;
7 }
8
9 int main(void) // main関数
10 {
11     printf("私は小野です.\n");
12     return 0;
13 }
```

出力は以下の通りである。

```
ono-sans@PC:~/ProgramEx01/w01$ ./ex01_03.out
Hello.
```

図4: 実行結果

図2のような結果を期待していたため、予想通りにはならなかった。

ここから考えられることとして恐らくコンパイラ側は機械語に翻訳する際 main 関数の中身のみコンパイルしているもしくはその部分のみ実行するようにしていると予想される。

test 関数のような main 関数以外の関数を使用する場合は何か他の方法があるのだろうと考えられる。

7 タイピング速度測定

これが私の記録です。



図5: 寿司打の記録

8 所感

初めてのプログラミング演習ということで初心に立ち返って授業を聞きましたが、意外と抜けていた部分などもあったためとても勉強になりました。レポートについても授業内で扱ったこと以外の知識はなるべく使わず、新しいことをする場合は信頼性の高いサイトからの情報を使って、「自分がここまでの知識しかもっていなかったとき、どのように考えるか」ということも考えながら考察を書きました。

参考文献

- [1] "Standard C++ Library Reference(日本語翻訳版)". IBM. ver.3.1.0. <https://www.ibm.com/docs/ja/zos/3.1.0?topic=functions-main-function>, (2024 年 10 月 7 日).