

2017 年 5 月 29 日

制作者：福永、井上

C 言語演習問題

全体を通しての注意

- 必ず作ったプログラムには、ソース内に製作日、名前、どのようなプログラムなのか等、適切なコメントをつけること。
- プログラム中には、想定外の入力があつた場合に対してのエラー処理も書いておくこと。
- 文言や意味、プログラムの書き方が分からない場合には、この演習を担当しているメンター(福永、井上)に質問すること。
- プログラムのファイル名は `problemX.c` (X は問題番号) とする。
- 6/14(水) 13:00 までに `gitlab` で提出すること。

問1 C コンパイラと Emacs を導入し、Welcome Yokolab!! を表示させる。

なお、C コンパイラがすでに導入済みである場合は、導入しなくても良い。
また、Vim などの他のエディタを使用しても良い。

問2 入力に対し 3 の倍数のとき、「3 の倍数です」と表示する。それ以外は「3 の倍数ではない」と表示する。入力は `scanf` 関数を使い取得する。

```
$ ./problem2
```

```
6
```

```
3 の倍数です
```

問3 入力に対し 2 の倍数ととき、「2 の倍数です」と表示する。3 の倍数のとき、「3 の倍数です」と表示する。それ以外の場合は「2 の倍数でも、3 の倍数でもない」と表示する。条件分岐は 2 の倍数のほうを優先すること。入力は `scanf` 関数を使い取得する。

```
$ ./problem3
```

```
6
```

```
2 の倍数です
```

```
$ ./problem3
```

```
11
```

```
2 の倍数でも 3 の倍数でもありません
```

問4 入力 `x` が素数かどうかを判定するプログラムを作成すること。素数である場

合は、「xは素数です」と表示する。素数でない場合は、「xは素数ではありません」と表示する。ループ処理は **for** 文のみで作成すること。入力は **scanf** 関数を使い取得する。

```
$ ./problem4
```

```
7
```

```
7は素数です
```

問5 九九の答えを表示するプログラムを作成すること。縦 9 列、横 9 列で表示すること。 **for** 文のみで作成すること。

問6 問 4 を改変し **for** 文を **while** 文に書き直すこと。

問7 “Lab name is Yokoyama Lab”と表示すること。長さ 20 の文字配列を作成し、その配列の中に文字列”Yokoyama Lab”を格納する。この文字配列に格納された文字列を利用すること。

問8 **int** 型配列の中に 1～10 までの数字を格納する。その後、その配列を用いて格納した数字を表示させる。

問9 二次元配列を用いて九九の答えを格納するプログラムを作成する。表示する内容はランダムで決められた、1~9 までの段のどれか 1 行を出力する。

```
$ ./problem9
```

```
2 4 6 ... 18
```

問10 1 から 10 までの数字を 4 で割る。このとき、余りが 2 のとき「Yokolab」、余りが 3 のとき「Member」、それ以外のときはもとの数字を表示する。(if else 文を使用せず **switch** 文を使用すること)

```
$ ./problem10
```

```
1, Yokolab, Member, 4, 5, Yokolab, Member, 8, 9, Yokolab
```

問11 コマンドライン引数から任意の数字を入力し、素数かどうかを判定するプログラムを作成すること。

```
$ ./problem11 7
```

```
7は素数です
```

問12 定数 x に 1.432645、定数 y に 1.567356 を **double** 型で定義すること。こ

の x と y を足してできた数、 a を用いて、`int` 型にキャストする前と、キャストした後をそれぞれ `printf` で表示し、キャストによって小数点以下が切り捨てられることを確認する。

問13 `int` 型の配列に 1~5 までの数字を格納する。その `int` 型に格納されている各要素の値と、その要素のアドレスを表示させること。

```
$ ./problem13
value: 1, address: 0x????????
value: 2, address: 0x????????
value: 3, ...
```

問14 `char` 型の配列に「12a45」を文字列として格納する。その文字列が格納している文字を 1 文字ずつ表示し、その文字ごとのアドレスをそれぞれ表示すること。

```
$ ./problem14
value: 1, address: 0x????????
value: 2, address: 0x????????
value: a, ...
```

問15 `int data1` と `int *data2` を利用して以下の処理を行うプログラムを作成すること。

1. 変数 `data1` に数値 5 を格納する
2. ポインタ変数 `data2` に変数 `data1` のアドレスを格納する
3. 変数 `data1` の値を出力する
4. ポインタ変数 `data2` を用いて `data1` の値を出力する
5. 変数 `data1` のアドレスを出力する
6. ポインタ変数 `data2` が格納している値を出力する

問16 自分の名前と年齢を格納する構造体を作成する。実行時のコマンドライン引数から、自分の名前と年齢を受け取り構造体に格納する。その後、構造体から、自分の名前と年齢を取り出し表示する。

問17 入力された数字に対して以下の表示を行うプログラムを作成する。

1. 5 の倍数のときに「yoko」と表示する
2. 7 の倍数のときに「lab」と表示する
3. 5 の倍数かつ 7 の倍数のときは「yokolab」と表示する。これは 1 と 2

の条件より優先される。

4. 1 と 2 と 3 の条件を満たさない素数のときは、16 進数で値を表示する。
5. それ以外は 10 進数で表示する。

入力のコマンドライン引数から取得すること。

素数判定は `main` 関数とは別に自作関数を作り、その関数を `main` 関数で用いること。

問18 `main` 関数内で `void counter1` 関数を 5 回呼び出す、その後、グローバル変数に入っている値を表示する。グローバル変数の初期値は `int` 型で 0 とする。

`counter1` 関数内では静的ローカル変数とローカル変数を `int` 型で 0 を定義する。その後、関数が呼び出されるたびに、グローバル変数、静的ローカル変数、ローカル変数それぞれのカウンターを 1 回進め、静的ローカル変数とローカル変数、グローバル変数をそれぞれ表示する。

問19 次に示す”`template.txt`”ファイルを読み出す(`fgets` 関数を用いること)。
`scanf` 関数を用いて、「あなたの名前は？」と表示した後に名前を入力させる。その後、「あなたの年齢は？」と表示し、年齢を入力させる。その後、`template.txt` から読みだした研究室名→名前→改行→年齢の順で `result.txt` に対して書き出しを行う(`fputs` 関数を用いること)。名前などをローマ字にして英数字のみで読み書きしてもよい。

```
template.txt -----  
Yokoyama Lab  
-----
```

問20 問 19 で使用した”`template.txt`”ファイルを読み出す(`fread` 関数を用いること)。`scanf` 関数を用いて、「あなたの名前は？」と表示した後に名前を入力するようにする。その後、「あなたの年齢は？」と表示し、年齢を入力するようにする。その後、`template.txt` から読みだした研究室名→名前→改行→年齢の順で `result.txt` に対して書き出しを行う(改行には `fput` 関数、それ以外には `fwrite` 関数を用いること)。名前などをローマ字にして英数字のみで読み書きしてもよい。