2023 Spring “Computer Programming”

Prof. Eom, Hyeonsang / T.A. An, Byoungchul

Exercise 0. <C programming>

- Submit the following file to “Assignments > Exercise 0” section in eTL.
  - **Zip file** (*.zip): The zip file must **contain the two types of file** below. The name must be “StudentID-EX0.zip”. No Hyphen-Minus (-) between student ID.

    e.g. **202320131-EX0.zip**
  - **Source Codes** (*.c): Each C file has the solution for the corresponding problem. The file name must be “StudentID-ProblemNumber”. No Hyphen-Minus (-) between student ID. Yes, that special character next to ‘0’ and used for subtractions.

    e.g. **202320131-1.c / 2020320131-2.c**
  - **Word file** (*.doc or *.docx): A single word file with the image of each expected output.

    e.g. You can find the example on the last page of this guide.

    The file name must be “StudentID-output.doc or .docx. No Hyphen-Minus (-) between student ID.

    e.g. **202320131-output.docx**

- If you could not solve a problem,
  - Type “**Could not solve**” at the first line of your source code **as a comment.**
  - Type “**Could not solve**” on the corresponding problem in your Word file.

    e.g. You can find the example on the last page of this guide.

- Specific IDEs or systems are **not** needed.
  - You can use any IDEs (e.g. Dev C++, VSCode, etc) or systems (Windows 10, Ubuntu, MacOS) for this exercise. You can use online complier sites if you want

- A skeleton code for Problem 4 is provided.
- **Modify only the parts with "..."** in the skeleton code to print the expected output.

1. **Print a simple string**

   - Print "Hello World!" in 'main' function.
   - Use 'printf()' to print the string.

   [Expected Output]

```
root@g414server:/home/g414/
Hello World
root@g414server:/home/g414/
```

2. **Print the two and three times table using while or for loop**

   - Print the two and three times table using loops in 'main' function.
   - Use 'printf()' to print the string.
   - Use loop counters to calculate and print all the integer values in the tables.
   - Use horizontal escape sequence (₩t) to separate between tables.

   [Expected Output]

```
root@g414server:/home/g414/ABC/
2 x 1 = 2         3 x 1 = 3
2 x 2 = 4         3 x 2 = 6
2 x 3 = 6         3 x 3 = 9
2 x 4 = 8         3 x 4 = 12
2 x 5 = 10        3 x 5 = 15
2 x 6 = 12        3 x 6 = 18
2 x 7 = 14        3 x 7 = 21
2 x 8 = 16        3 x 8 = 24
2 x 9 = 18        3 x 9 = 27
root@g414server:/home/g414/ABC/
```

3. **Print integer value 100 ~ 109 using array and index using a function**

   - Define a macro (identifier = MAX_ARR_SIZE, token-string = 10) with "#define" directive.
   - Define a function called "printArray()" that takes an integer pointer type as a parameter named "arr" and returns nothing.
   - Declare a prototype of the function right under the macro.
   - Allocate a local array named 'MyArr' with macro (MAX_ARR_SIZE) in 'main' function
   - Assign integer values (100 ~ 109) to the array using a for loop.
   - Print the value stored in each element of the array using "printArray()" function.
   - Print new line sequence (₩n) after printing the last integer.

   [Expected Output]

```
root@g414server:/home/g414/ABC/Personal_stu
100 101 102 103 104 105 106 107 108 109
root@g414server:/home/g414/ABC/Personal_stu
```

4. Print the name of vehicles and number of wheels

- Define a user-defined 'struct' data type called 'myVehicle' with 'typedef' keyword
- 'myVehicle' has 4 fields:
  - name: an array that can hold 10 characters
  - numOfWheels: an integer value
  - Setup(): a function pointer of a function that takes a pointer of 'myVehicle' data type, a character pointer, and an integer as parameters and returns nothing.
  - Stat(): a function pointer of a function that takes a pointer of 'myVehicle' data type as a parameter and returns nothing.
- Define a function called "mySetup" that takes a pointer of 'myVehicle' data type (called 'pv'), a character pointer (called 'name'), and an integer (called 'n') as parameters and returns nothing.
  - The function assigns the string that the character pointer 'name' points to 'name' field of 'pv'
  - The function prints "Setup: 'name' with 'n' wheels" using the arguments passed.
  - The function assigns the integer value 'n' to 'numOfWheels' field of 'pv'
- Define a function called "myStat" that takes a pointer of 'myVehicle' data type (called 'pv') as a parameter.
  - The function prints "Stat: 'name' has 'n' wheels" using the arguments passed.
- Define a function called "initVehicle" that takes a pointer of 'myVehicle' data type (called 'pv') as a parameter.
  - The function assigns 'mySetup' function to 'Setup' field of 'pv' and 'myStat' function to .'Stat' field of 'pv'
- Use 'initVehicle' function, 'Setup' and 'Stat' field of 'myVehicle' data type in 'main' function to print the desired output.

  [Expected Output]

```
root@g414server:/home/g414/
Setup: Bus with 4 wheels
Stat: Bus has 4 wheels
Setup: Bike with 2 wheels
Stat: Bike has 2 wheels
root@g414server:/home/g414/
```

[Source Code Example]

Name: 202320131-1.c

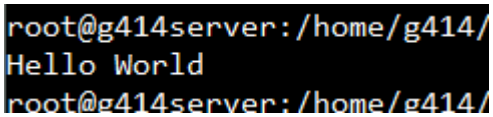
```
1  #include <stdio.h>
2
3  int main() {
4          ... // Your Solution
5
6
```

Name: 202320131-3.c

```
1  # Could not solve
2  #include <something>
3
4  int something // Attempts to solve the problem
5
6
```

[Word file Example]

Name: 202320131-output.docx

1. 

2. Could not solve
3. Could not solve
4.