Creative Software Design, Assignment 2-1

Handed out: Sep 8, 2021

Due: 23:59, Sep 8, 2021 **(NO SCORE for late submissions!)**

- *Only files submitted by **git push to this course project at https://hconnect.hanyang.ac.kr** (<Year>_<Course no.>_<Class code>/<Year>_<Course no.>_<Student ID>.git) will be scored.*
- *Place your files under the directory structure **<Assignment name>/<Problem no.>/<your files>** just like the following example.*

```
+ 2020_ITE0000_2019000001
  + 2-1/
    + 1/
      - 1.c
      - Makefile
    + 2/
      - 2.c
      - Makefile
    + …
```

- *The submission time is determined not when the commit is made **but when the git push is made**.*
- *Your files must be committed to the **master branch**. Otherwise, it will not be scored.*
- *Your program should output correct results even for inputs other than those used in the example.*
- *Basically, assignments are scored based on the output results. If it is not possible to check whether a requirement is implemented because the output is not correct, no score is given for the requirement, even if it is implemented internally. However, even if the output result is correct, no score is given for a requirement if the internal implementation does not satisfy the requirement.*

1. Write a program that works as follows.

   A. Define a structure named Book that can store the name and published year of a book.

B. Create a Book type array of length 4.

C. Take four names and published years from the user and store them in each element of the array.

  i. Assume that each name is one English word, less than 20 letters, and each published year is an integer number larger than 0.

D. Print out the contents of the array.

E. Input: Four pairs of name and published year

F. Output: The stored name and published year in the array

G. Files to submit:

  i. A C source file

  ii. A Makefile to generate the executable

```
$ ./array_struct_book
Cosmos 1980⏎
TheOrderOfTime 2017⏎
AnimalFarm 1945⏎
Gold 2015⏎
Name: Cosmos, Published Year: 1980
Name: TheOrderOfTime, Published Year: 2017
Name: AnimalFarm, Published Year: 1945
Name: Gold, Published Year: 2015
```

2. Complete the code skeleton below to write a program to manipulate a 2D point.

A. Create two variables (p1 and p2) of Point type (see the code skeleton below) and fill in the value of each member by taking two integers(total four) from user.

B. Then call getScale2xPoint() (see the code skeleton below) to make each Point twice as large as that of the original.

C. Print out the value of p1 and p2.

D. Swap the value of p1 and p2 using swapPoints() (see the code skeleton below).

E. Print out the value of p1 and p2 again.

F. Note that

  i. The code for printing Point variables must be in main().

G. Input: Two integer numbers

H. Output: The value of p1 & p2 after calling getScale2xPoint() and swapPoints()

I. Files to submit:

   i. A C source file

   ii. A Makefile to generate the executable

```
$./point
1 2
2 5
Calling getScale2xPoint()
P1 : 2 4
P2 : 4 10
Calling swapPoint()
P1 : 4 10
P2 : 2 4
$
```

Code skeleton:

```
typedef struct
{
        int xpos;
        int ypos;
}Point;

Point getScale2xPoint(const Point* p){
        //Implement this function
}

void swapPoints(Point* p1, Point* p2){
        //Implement this function
}

int main(void){
        //Implement this function
}
```