<Advanced C Programming and Lab> Ch 11. Structures

※ Note

- If not mentioned, assume that there is no additional inputs.

- If not mentioned, do not print a space in the beginning and end of each line.

- In input and output examples, after $\mapsto$ symbol is to explain the input and output.

- In output examples, □ symbol indicates a space.

Section2 [Problem 1] Define a structure to store a 3-dimensional vector (x, y, z) and declare 2 3-dimensional vectors (V_1, V_2) and receive their coordinates. Add two vectors and print the result.

   - vectors are int data type

Input Example 1	Output Example 1
1 2 3 　　　 $\mapsto$ V_1 -1 5 5 　　　 $\mapsto$ V_2	0 7 8 　　　 $\mapsto$ $V_1 + V_2$

Section2 [Problem 2] Define a structure to store a 3-dimensional vector (x, y, z) and declare 2 3-dimensional vectors (V_1, V_2) and receive their coordinates. Declare a new vector (V_3), containing the product of V_1 and V_2, and print the result. Compute the inner product of V_1 and V_2, and print the result.

   - vectors are int data type
   - product of two 3-dimensional vectors V_1={ x_1, y_1, z_1 } and V_2={ x_2, y_2, z_2 }:
     V_3 = {x_1 x x_2, y_1 x y_2, z_1 x z_2}
   - Inner product of two 3-dimensional vectors V_1={ x_1, y_1, z_1 } and V_2={ x_2, y_2, z_2 }:
     $V_1 \cdot V_2$ = $x_1 x_2 + y_1 y_2 + z_1 z_2$

Input Example 1	Output Example 1
1 2 3 　　　 $\mapsto$ V_1 -1 5 5 　　　 $\mapsto$ V_2	-1 10 15 　　　 $\mapsto$ V_3 24 　　　　　　 $\mapsto$ Inner product

Section2 [Problem 3] Use structures to show the current time. Receive 2 different time points and compute and print the time difference.

Input Example 1	Output Example 1
10 20 30 ↦ 10h 20m 30s 12 05 10 ↦ 12h 05m 10s	1 44 40

Input Example 2	Output Example 2
1 10 20 ↦ 1h 10m 20s 3 20 30 ↦ 3h 20m 30s	2 10 10

Section2 [Problem 4] Define a structure "fraction", compute the product of two fractional numbers.

- Structure fraction has two members: int type numerator and denominator
- Ex) product of two fractional numbers: 4/5 × 3/7 = 12/35

Input Example 1	Output Example 1
4 5 3 7	12/35

Input Example 2	Output Example 2
2 10 51 22 ↦ 2/10 × 51/22 = 102/220	102/220

Section3 [Problem 5] Receive 5 students' name and final exam score and store them in a structure array. Print the name of the students whose final exam score is less than the average score

- Student's name: 9 alphabet letters without spaces.

Input Example 1	Output Example 1
akim 75 bkim 85 ckim 65 dkim 95 ekim 100	akim ckim

Section3 [Problem 6] Receive N students' name and 3 exam scores. Compute and print each student's name, aveage score (1 decimal point), and grade.

- Receive N first: number of students, N is less than or equal to 20.
- Students' name: 19 alphabet letters without spaces.

- Grade: A if average score≥90 and ≤100, B if average score≥80 and <90, C if average score≥70 and <80, otherwise F

Input Example 1	Output Example 1
1 → N Hongildong 95 80 75	Hongildong 83.3 B

Section3 [Problem 7] Receive 10 integers. Compute the rank of the integers in descending order. Print the 3rd and 7th ranked integers.

  - No overlapping integers.

  - Do not sort and store the integers in an array. Store the integers as they are received in an array.

Input Example 1	Output Example 1
78 65 23 43 82 95 31 15 8 54	65 23

Section3 [Problem 8] Receive information of N students (name, scores: Korean, English, Math). Compute the grade of each student based on the average score. Print each student's name, average score (1 decimal point), and grade.

  - N: ≥1 and ≤50.   - name: alphabet letters without spaces

  - name: alphabet letters (1 to 8) without spaces

  - scores range from 0 to 100

  - average: 1 decimal point

  - Grade: A if ≥90 and ≤100, B if ≥80 and <90, C if ≥70 and <80, otherwise D

  - Cannot use global variables

  - Must use structure pointers

  - Define and use the following student structure

      a character type array to store names, int type arrays to store Korean, English, Math scores, a double type array to store average scores, a character type array to store grades

Input Example 1	Output Example 1
2 Kim 82 72 58 Young 90 100 99	Kim□70.7□C Young□96.3□A

Section5 [Problem 9] Receive two complex numbers, print the sum of the two complex numbers.

- Cannot use global variables
- Define and use a structure complex to represent complex numbers
  - ○ double type variables to represent real and imaginary parts
- Define and use a function add
  - ○ Arguments: two complex structure variables
  - ○ Return type: complex structure variable
- main ()
  - ○ Declare complex structures to store complex numbers
  - ○ Receive two complex numbers from a user
  - ○ Call add(), compute the sum of two complex numbers
  - ○ Print the sum

Input Example 1	Output Example 1
2.3 4.5 ↦ complex number (=2.3+4.5i) 3.4 5.0 ↦ complex number (=3.4+5i)	5.7 + 9.5i ↦ complex number: sum

Section5 [Problem 10] Receive one student's exam results (maximum score, minimum score, cutoff value). Print pass/fail (P or F) and the score range.
- Pass/fail: Pass if difference between the maximum and minimum scores is less than or equal to the cutoff value!
- Maximum and minimum scores range from 0 to 100
- If pass, print 'P'. If fail, print 'F'
- Cutoff value (int type) ranges from 0 to 15.
- Define and use a structure result to store the exam results and informaion
  - ○ int type variables: maximum score, minimum score, cutoff value, score range. character variable: pass/fail
- Cannot use global variables
- Define and use a function passorfail()
  - ○ arguments: structure pointers
  - ○ store 'P' or 'F' (pass or fail)
  - ○ return type: none
- main()
  - ○ Declare a structure result
  - ○ Receive the maximum score, minimum score, and cutoff value from a user

○ Call passorfail(), determine if pass or fail

○ Print score range and 'P' or 'F'

Input Example 1		Output Example 1
100 95 10 ↦ maximum, minimum, cutoff value		5 P

[Problem 11] Receive the number of students in a class. Receive students' sex, weight, height. Print the body grade according to the following table.

Male(1)

	Height<165	165≤Height<175	175≤Height
Weight<60	1	2	3
60≤Weight<70	3	1	2
70≤Weight	2	3	1

Female(2)

	Height<165	165≤Height<175	175≤Height
Weight<50	1	2	3
50≤Weight<60	3	1	2
60≤Weight	2	3	1

- Sex: Male – 1, Female – 2

- Use a structure to store students' data

- Number of students≤10

Input Example 1	Output Example 1
2 ↦ Number of students	2 0 0 ↦ 1st, 2nd, 3rd grade
1 66 170 ↦ sex, weight, height	
2 48 155 ↦ sex, weight, height	

[Problem 12] Receive 10 students' name and 3 quiz scores. Print the name and average score of the students who obtained the highest and lowest average scores. Print the name and average score of the students who belong to the bottom 30% based on the average score. Use the following functions.

- read_data( )

○ arguments: structure arrays

○ receive 10 students' name and 3 quiz scores and store them in a structure array

○ student's name: ≤9 alphabet letters without spaces

- ○ return type: none
- cal_avg( )
  - ○ arguments: structure arrays
  - ○ compute each student's average quiz score
  - ○ return type: none
- sort( )
  - ○ arguments: structure arrays
  - ○ Sort the students according to their average score in descending order
  - ○ return type: none
- print_score( )
  - ○ arguments: structure arrays
  - ○ print students' name and average score who obtained the maximum average score, minimum average score, and bottom 30% average score. Print the average score using 2 decimal points
  - ○ return type: none
- No two students have the same average score

Input Example 1	Output Example 1	
A.Kim 80 80 80	B.Kim□90.00	↦ maximum average score
B.Kim 90 90 90	A.Kim 80.00	↦ minimum average score
C.Kim 81 81 81	A.Lee 82.00	↦ bottom 30% (descending order)
A.Lee 82 82 82	C.Kim 81.00	↦ bottom 30% (descending order)
B.Lee 83 83 83	A.Kim 80.00	↦ bottom 30% (descending order)
C.Lee 84 84 84		
A.Park 85 85 85		
B.Park 86 86 86		
C.Park 87 87 87		
A.Choi 88 88 88		