프로그래밍언어 (실습)

실습 9 (보충설명) – 구조체, 구조체 배열, 구조체 배열의 통계분석 및 정렬

교수 김 영 탁

영남대학교 기계IT대학 정보통신공학과

(Tel : +82-53-810-2497; E-mail : ytkim@yu.ac.kr)

Outline

- **구조체 설계, 구현: Date, Tel_Number, Student**
  - 구조체 출력: printDate(), printTelNumber(), printStudent()
  - 구조체 변수의 비교: compareDate(), compareTelNumber()
- **구조체 Student의 배열의 통계 분석, 탐색, 정렬 기능 구현**
  - void statisticsGPA(Student students[], int num_students);
  - Student *searchBestGPAStudent(Student students[], int num);
  - Student *searchWorstGPAStudent(Student students[], int num);
  - void sortStudents_by_GPA(Student students[], int num);
  - void sortStudents_by_ST_ID(Student students[], int num);
  - void sortStudents_by_name(Student students[], int num);
  - void sortStudents_by_BirthDate(Student students[], int num);
  - void sortStudents_by_TelNumber(Student students[], int num);

학생을 위한 구조체 (Student) 선언
Student 구조체 배열과 통계처리
Student 구조체 배열의 탐색과 정렬

구조체 Date 구현

```
/* Date.h */
#ifndef DATE_H
#define DATE_H
typedef struct
{
        int year;
        int month;
        int day;
} Date;

void printDate(Date date);
int compareDate(Date d1, Date d2);
#endif
```

```
/* Date.cpp */
#include <stdio.h>
#include "Date.h"

void printDate(Date date)
{
        printf("(%04d, %2d, %2d)",
            date.year, date.month, date.day);
}

int compareDate(Date d1, Date d2)
{
        if (d1.year > d2.year)
                return 1;
        else if (d1.year < d2.year)
                . . . . // 직접 구현할 것
        else
                return 0;
}
```

구조체 Tel_Number의 구현 (1)

```
/* Telephone_Number.h */

#ifndef TELEPHONE_NUMBER_H
#define TELEPHONE_NUMBER_H
#define U_SHORT unsigned short
typedef struct
{
        U_SHORT nation_code;
        U_SHORT region_no;
        U_SHORT switch_no;
        U_SHORT line_no;
} Tel_Number;

void printTelephoneNumber(Tel_Number telNo);
int compareTelNumber(Tel_Number tn1, Tel_Number tn2);
#endif
```

구조체 Tel_Number의 구현 (2)

```
/* Telephone_Number.cpp */
#include <stdio.h>
#include "Telephone_number.h"

void printTelephoneNumber(Tel_Number telNo)
{
        printf("(tel: +%d-", telNo.nation_code);
        printf("%03d-%04d-%04d)", telNo.region_no, telNo.switch_no, telNo.line_no);
}

int compareTelNumber(Tel_Number tn1, Tel_Number tn2)
{
        if (tn1.nation_code > tn2.nation_code)
                return 1;
        else if (tn1.nation_code < tn2.nation_code)
                return -1;
        else if (tn1.region_no > tn2.region_no)
                . . . . . // 직접 구현할 것
        else
                return 0;
}
```

구조체 Student의 구현

```
/* Student.h  (1) */
#ifndef STUDENT_H
#define STUDENT_H

#include "Date.h"
#include "Telephone_number.h"
#define MAX_NAME_LEN 20
#define NUM_STUDENTS 10

typedef struct
{
        int st_id;
        char name[MAX_NAME_LEN];
        Date birth_date;
        Tel_Number tel_number;
        double GPA; // Grade Point Average
} Student;
```

```
/* Student.h  (2) */

void printStudent(Student *pST);
void printStudents(Student *stArr, int num_students);
void shuffleStudents(Student students[], int num_students);
void statisticsGPA(Student students[], int num_students);
Student *searchBestGPAStudent(Student students[], int num);
Student *searchWorstGPAStudent(Student students[], int num);
void sortStudents_by_GPA(Student students[], int num);
void sortStudents_by_ST_ID(Student students[], int num);
void sortStudents_by_name(Student students[], int num);
void sortStudents_by_BirthDate(Student students[], int num);
void sortStudents_by_TelNumber(Student students[], int num);

#endif
```

```
/* Student.cpp  (1) */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include "Student.h"
#include "Date.h"
#include "Telephone_number.h"

void printStudent(Student *pST)
{
	printf("Student [ID: %08d, %-10s", pST->st_id, pST->name);
	printf(", GPA: %5.2lf", pST->GPA);
	printf(", ");
	printDate(pST->birth_date);
	printf(", ");
	printTelephoneNumber(pST->tel_number);
	printf("]");
}

void printStudents(Student *stArr, int num)
{
	Student *st = stArr;
	for (int i = 0; i < num; i++)
	{
		printStudent(st);
		printf("\n");
		st++;
	}
}
```

```
/* Student.cpp  (2) */

void shuffleStudents(Student stArray[], int num_students)
{
      int st_1, st_2;
      Student st_temp;

      srand(time(0));
      for (int i = 0; i < num_students; i++)
      {
            st_1 = rand() % num_students;
            st_2 = rand() % num_students;

            . . . .  // swap stArray[st_1] and stArray[st_2];
      }
}

void statisticsGPA(Student students[], int num_students)
{
      /* Calculate the statistics of students' GPA : max, min, avg */
      Student *pST, *pST_max, *pST_min;
      pST_max = pST_min = pST = students;
      double GPA_max, GPA_min, GPA_avg, GPA_sum = 0.0;
      GPA_max = GPA_min =  pST->GPA;
      GPA_sum = pST->GPA;
      for (int i = 1; i < num_students; i++)
      {
            pST = &students[i];
            . . . . .  // determine new GPA_max and GPA_min
            GPA_sum += pST->GPA;
      }  // end for
      GPA_avg = GPA_sum / (double) num_students;
      printf("GPA_max (%5.2lf), GPA_min (%5.2lf), GPA_avg (%5.2lf)\n",
          GPA_max, GPA_min, GPA_avg);
}
```

```
/* Student.cpp  (3) */

Student *searchBestGPAStudent(Student students[], int num)
{
      /* Search the student with highest GPA */
      Student *pST, *pST_max;
      pST_max = pST = students;
      double GPA_max;
      GPA_max = pST->GPA;
      for (int i = 1; i < num; i++)
      {
            pST = &students[i];
            . . . .  // check whether new GPA_max is found
      }
      return pST_max;
}

Student *searchWorstGPAStudent(Student students[], int num)
{
      /* Search the student with highest GPA */
      Student *pST, *pST_min;
      pST_min = pST = students;
      double GPA_min;
      GPA_min = pST->GPA;
      for (int i = 1; i < num; i++)
      {
            pST = &students[i];
            . . . . // check whether new GPA_min is found
      }

      return pST_min;
}
```

```
/* Student.cpp  (3) */

void sortStudents_by_GPA(Student students[], int num)
{
      /* Selection Sorting student array by GPA in decreasing order */
      Student *pST, *pST_max;
      Student temp;
      int st_max_id;

      for (int i = 0; i < num; i++)
      {
            pST_max =  pST = &students[i];
            st_max_id = i;
            for (int j = i + 1; j < num; j++)
            {
                  pST = &students[j];
                  . . . .  // check whether new GPA_max is found
            }  // end for
            if (st_max_id != i)
            {
                  . . . .  // swap students[i] and students[st_max_id];
            }
      }

}
```

```
/* Student.cpp  (4) */

void sortStudents_by_ST_ID(Student students[], int num)
{
      /* Selection Sorting student array by ST_ID in increasing order */
      Student *pST, *pST_min;
      Student temp;
      int st_id_min;

      for (int i = 0; i < num; i++)
      {
            pST_min = pST = &students[i];
            st_id_min = i;
            for (int j = i + 1; j < num; j++)
            {
                  pST = &students[j];

                  . . . . // check whether new st_id_min is found
            }  // end for
            if (st_id_min != i)
            {
                  . . . .  // swap students[i] and students[st_in_min];
            }
      }

}
```

```
/* Student.cpp  (5) */

void sortStudents_by_name(Student students[], int num)
{
	/* Selection Sorting student array by name in increasing order */
	Student *pST, *pST_min;
	Student temp;
	int st_name_min;

	for (int i = 0; i < num; i++)
	{
		pST_min = pST = &students[i];
		st_name_min = i;
		for (int j = i + 1; j < num; j++)
		{
			pST = &students[j];

			if (strcmp(pST->name, pST_min->name) < 0)
			{
				pST_min = pST;
				st_name_min = j;
			}
		} // end inner for
		if (st_name_min != i)
		{
			. . . .  // swap students[i] and students[st_name_min];
		}
	} // end outer for

}
```

```
/* Student.cpp  (5) */

void sortStudents_by_BirthDate(Student students[], int num)
{
	/* Selection Sorting student array by BirthDate in increasing order */
	Student *pST, *pST_min;
	Student temp;
	int st_date_min;

	for (int i = 0; i < num; i++)
	{
		pST_min = pST = &students[i];
		st_date_min = i;
		for (int j = i + 1; j < num; j++)
		{
			pST = &students[j];

			if (compareDate(pST->birth_date, pST_min->birth_date) < 0)
			{
				pST_min = pST;
				st_date_min = j;
			}
		}
		if (st_date_min != i)
		{
			. . . .  // swap students[i] and students[st_date_min];
		}
	}

}
```

```
/* Student.cpp  (5) */

void sortStudents_by_TelNumber(Student students[], int num)
{
	/* Selection Sorting student array by TelNo in increasing order */
	Student *pST, *pST_min;
	Student temp;
	int st_telno_min;

	for (int i = 0; i < num; i++)
	{
		pST_min = pST = &students[i];
		st_telno_min = i;
		for (int j = i + 1; j < num; j++)
		{
			pST = &students[j];

			if (compareTelNumber(pST->tel_number, pST_min->tel_number) < 0)
			{
				pST_min = pST;
				st_telno_min = j;
			}
		}
		if (st_telno_min != i)
		{
			. . . .  // swap students[i] and students[st_telno_min];
		}
	}

}
```

Student 구조체 배열
기능 시험 Program

StudentRecords.cpp

```
/* StudentRecords */
#include "Student.h"

Student students[] =
{
        { 21911000, "Kim, G-M", { 1990, 10, 5 }, { 82, 53, 805, 1234 }, 3.57 },
        { 21822075, "Yoon, S-M", { 1990, 4, 5 }, { 82, 53, 811, 1550 }, 4.37 },
        { 21433015, "Hwang, S-S", { 1989, 1, 10 }, { 82, 53, 817, 1005 }, 2.72 },
        { 21644054, "Lee, K-M", { 1991, 5, 15 }, { 82, 10, 9112, 9876 }, 3.35 },
        { 21255340, "Hong, G-M", { 1990, 2, 5 }, { 82, 55, 810, 5678 }, 3.57 },
        { 21766056, "Zang, S-M", { 1990, 3, 15 }, { 82, 10, 9112, 1600 }, 4.45 },
        { 21177017, "Park, S-S", { 1989, 7, 10 }, { 82, 34, 817, 1098 }, 4.12 },
        { 21588053, "Choi, Y-H", { 1992, 9, 25 }, { 82, 53, 845, 5764 }, 3.85 },
        { 21399019, "Ahn, D-J", { 1988, 10, 3 }, { 82, 31, 817, 1038 }, 3.21 },
        { 21010053, "Kwak, S-B", { 1994, 11, 15 }, { 82, 2, 897, 8778 }, 4.43 },
        { 21021053, "Song, W-B", { 1993, 5, 15 }, { 82, 34, 345, 1234 }, 4.47 },
        { 21133053, "Gong, G-W", { 1992, 6, 25 }, { 82, 55, 543, 4321 }, 2.35 },
        { 20911042, "Bhang, S-H", { 1987, 12, 25 }, { 82, 2, 123, 4567 }, 3.75 },
        {-1}
};
```

main()

```
/* Example of Handling an Array of Struct Student (1)  */

#include <stdio.h>
#include "Student.h"
#define MAX_NUM_STUDENTS 100

void main()
{
	int num_students = 0;
	int menu;

	Student *pST_GPA_max, *pST_GPA_min;
	extern Student students[];
	for (int i = 0; i < MAX_NUM_STUDENTS; i++)
	{
		if (students[i].st_id == -1)
			break;
		else
			num_students++;
	}
	printf("Number of students = %d\n", num_students);
```

```
/* Example of Handling an Array of Struct Student (2)  */

      while (1)
      {
            printf("\n");
            printf(" 1 : print student records\n");
            printf(" 2 : calculate statistics GPA (max, min, avg) of students' \n");
            printf(" 3 : search students of best GPA and worst GPA\n");
            printf(" 4 : sort students by student ID\n");
            printf(" 5 : sort students by GPA\n");
            printf(" 6 : sort students by name\n");
            printf(" 7 : sort students by date of birth\n");
            printf(" 8 : sort studetns by telephone number\n");

            printf("0 : Quit\n");

            printf("Input menu = ");
            scanf("%d", &menu);
            if (menu == 0)
                  break;
```

/* Example of Handling an Array of Struct Student (3) */

```
        switch (menu)
        {
        case 1:
            printStudents(students, num_students);
            printf("\n");
            break;
        case 2:
            statisticsGPA(students, num_students);
            break;
```

```
Input menu = 1
Student [ID: 21911000, Kim, G-M  , GPA:  3.57, (1990, 10,  5), (tel: +82-053-0805-1234)]
Student [ID: 21822075, Yoon, S-M , GPA:  4.37, (1990,  4,  5), (tel: +82-053-0811-1550)]
Student [ID: 21433015, Hwang, S-S, GPA:  2.72, (1989,  1, 10), (tel: +82-053-0817-1005)]
Student [ID: 21644054, Lee, K-M  , GPA:  3.35, (1991,  5, 15), (tel: +82-010-9112-9876)]
Student [ID: 21255340, Hong, G-M , GPA:  3.57, (1990,  2,  5), (tel: +82-055-0810-5678)]
Student [ID: 21766056, Zang, S-M , GPA:  4.45, (1990,  3, 15), (tel: +82-010-9112-1600)]
Student [ID: 21177017, Park, S-S , GPA:  4.12, (1989,  7, 10), (tel: +82-034-0817-1098)]
Student [ID: 21588053, Choi, Y-H , GPA:  3.85, (1992,  9, 25), (tel: +82-053-0845-5764)]
Student [ID: 21399019, Ahn, D-J  , GPA:  3.21, (1988, 10,  3), (tel: +82-031-0817-1038)]
Student [ID: 21010053, Kwak, S-B , GPA:  4.43, (1994, 11, 15), (tel: +82-002-0897-8778)]
Student [ID: 21021053, Song, W-B , GPA:  4.47, (1993,  5, 15), (tel: +82-034-0345-1234)]
Student [ID: 21133053, Gong, G-W , GPA:  2.35, (1992,  6, 25), (tel: +82-055-0543-4321)]
Student [ID: 20911042, Bhang, S-H, GPA:  3.75, (1987, 12, 25), (tel: +82-002-0123-4567)]
```

```
Input menu = 2
GPA_max ( 4.47), GPA_min ( 2.35), GPA_avg ( 3.71)
```

```
/* Example of Handling an Array of Struct Student (3)  */

        case 3:
            pST_GPA_max = searchBestGPAStudent(students, num_students);
            pST_GPA_min = searchWorstGPAStudent(students, num_students);
            printf("Student with best GPA  : ");
            printStudent(pST_GPA_max);
            printf("\n");
            printf("Student with worst GPA : ");
            printStudent(pST_GPA_min);
            printf("\n");
            break;
```

```
Input menu = 3
Student with best GPA  : Student [ID: 21021053, Song, W-B , GPA:  4.47, (1993,  5, 15), (tel: +82-034-0345-1234)]
Student with worst GPA : Student [ID: 21133053, Gong, G-W , GPA:  2.35, (1992,  6, 25), (tel: +82-055-0543-4321)]
```

```
/* Example of Handling an Array of Struct Student (3)  */

        case 4:
            sortStudents_by_ST_ID(students, num_students);
            printf("After sorting students by increasing order of student ID:\n");
            printStudents(students, num_students);
            printf("\n");
            break;
```

```
Input menu = 4
After sorting students by increasing order of student ID:
Student [ID: 20911042, Bhang, S-H, GPA:  3.75, (1987, 12, 25), (tel: +82-002-0123-4567)]
Student [ID: 21010053, Kwak, S-B , GPA:  4.43, (1994, 11, 15), (tel: +82-002-0897-8778)]
Student [ID: 21021053, Song, W-B , GPA:  4.47, (1993,  5, 15), (tel: +82-034-0345-1234)]
Student [ID: 21133053, Gong, G-W , GPA:  2.35, (1992,  6, 25), (tel: +82-055-0543-4321)]
Student [ID: 21177017, Park, S-S , GPA:  4.12, (1989,  7, 10), (tel: +82-034-0817-1098)]
Student [ID: 21255340, Hong, G-M , GPA:  3.57, (1990,  2,  5), (tel: +82-055-0810-5678)]
Student [ID: 21399019, Ahn, D-J  , GPA:  3.21, (1988, 10,  3), (tel: +82-031-0817-1038)]
Student [ID: 21433015, Hwang, S-S, GPA:  2.72, (1989,  1, 10), (tel: +82-053-0817-1005)]
Student [ID: 21588053, Choi, Y-H , GPA:  3.85, (1992,  9, 25), (tel: +82-053-0845-5764)]
Student [ID: 21644054, Lee, K-M  , GPA:  3.35, (1991,  5, 15), (tel: +82-010-9112-9876)]
Student [ID: 21766056, Zang, S-M , GPA:  4.45, (1990,  3, 15), (tel: +82-010-9112-1600)]
Student [ID: 21822075, Yoon, S-M , GPA:  4.37, (1990,  4,  5), (tel: +82-053-0811-1550)]
Student [ID: 21911000, Kim, G-M  , GPA:  3.57, (1990, 10,  5), (tel: +82-053-0805-1234)]
```

```
/* Example of Handling an Array of Struct Student (3)  */

            case 5:
                  sortStudents_by_GPA(students, num_students);
                  printf("After sorting students by decreasing order of GPA:\n");
                  printStudents(students, num_students);
                  printf("\n");
                  break;
```

```
Input menu = 5
After sorting students by decreasing order of GPA:
Student [ID: 21021053, Song, W-B , GPA:  4.47, (1993,  5, 15), (tel: +82-034-0345-1234)]
Student [ID: 21766056, Zang, S-M , GPA:  4.45, (1990,  3, 15), (tel: +82-010-9112-1600)]
Student [ID: 21010053, Kwak, S-B , GPA:  4.43, (1994, 11, 15), (tel: +82-002-0897-8778)]
Student [ID: 21822075, Yoon, S-M , GPA:  4.37, (1990,  4,  5), (tel: +82-053-0811-1550)]
Student [ID: 21177017, Park, S-S , GPA:  4.12, (1989,  7, 10), (tel: +82-034-0817-1098)]
Student [ID: 21588053, Choi, Y-H , GPA:  3.85, (1992,  9, 25), (tel: +82-053-0845-5764)]
Student [ID: 20911042, Bhang, S-H, GPA:  3.75, (1987, 12, 25), (tel: +82-002-0123-4567)]
Student [ID: 21255340, Hong, G-M , GPA:  3.57, (1990,  2,  5), (tel: +82-055-0810-5678)]
Student [ID: 21911000, Kim, G-M  , GPA:  3.57, (1990, 10,  5), (tel: +82-053-0805-1234)]
Student [ID: 21644054, Lee, K-M  , GPA:  3.35, (1991,  5, 15), (tel: +82-010-9112-9876)]
Student [ID: 21399019, Ahn, D-J  , GPA:  3.21, (1988, 10,  3), (tel: +82-031-0817-1038)]
Student [ID: 21433015, Hwang, S-S, GPA:  2.72, (1989,  1, 10), (tel: +82-053-0817-1005)]
Student [ID: 21133053, Gong, G-W , GPA:  2.35, (1992,  6, 25), (tel: +82-055-0543-4321)]
```

```
/* Example of Handling an Array of Struct Student (3)  */

            case 6:
                sortStudents_by_name(students, num_students);
                printf("After sorting students by increasing order of student name:\n");
                printStudents(students, num_students);
                printf("\n");
                break;
```

```
Input menu = 6
After sorting students by increasing order of student name:
Student [ID: 21399019, Ahn, D-J  , GPA:  3.21, (1988, 10,  3), (tel: +82-031-0817-1038)]
Student [ID: 20911042, Bhang, S-H, GPA:  3.75, (1987, 12, 25), (tel: +82-002-0123-4567)]
Student [ID: 21588053, Choi, Y-H , GPA:  3.85, (1992,  9, 25), (tel: +82-053-0845-5764)]
Student [ID: 21133053, Gong, G-W , GPA:  2.35, (1992,  6, 25), (tel: +82-055-0543-4321)]
Student [ID: 21255340, Hong, G-M , GPA:  3.57, (1990,  2,  5), (tel: +82-055-0810-5678)]
Student [ID: 21433015, Hwang, S-S, GPA:  2.72, (1989,  1, 10), (tel: +82-053-0817-1005)]
Student [ID: 21911000, Kim, G-M  , GPA:  3.57, (1990, 10,  5), (tel: +82-053-0805-1234)]
Student [ID: 21010053, Kwak, S-B , GPA:  4.43, (1994, 11, 15), (tel: +82-002-0897-8778)]
Student [ID: 21644054, Lee, K-M  , GPA:  3.35, (1991,  5, 15), (tel: +82-010-9112-9876)]
Student [ID: 21177017, Park, S-S , GPA:  4.12, (1989,  7, 10), (tel: +82-034-0817-1098)]
Student [ID: 21021053, Song, W-B , GPA:  4.47, (1993,  5, 15), (tel: +82-034-0345-1234)]
Student [ID: 21822075, Yoon, S-M , GPA:  4.37, (1990,  4,  5), (tel: +82-053-0811-1550)]
Student [ID: 21766056, Zang, S-M , GPA:  4.45, (1990,  3, 15), (tel: +82-010-9112-1600)]
```

```
/* Example of Handling an Array of Struct Student (4)  */

            case 7:
                  sortStudents_by_BirthDate(students, num_students);
                  printf("After sorting students by increasing
                     order of student birth date:\n");
                  printStudents(students, num_students);
                  printf("\n");
                  break;
```

```
Input menu = 7
After sorting students by increasing order of student birth date:
Student [ID: 20911042, Bhang, S-H, GPA:  3.75, (1987, 12, 25), (tel: +82-002-0123-4567)]
Student [ID: 21399019, Ahn, D-J  , GPA:  3.21, (1988, 10,  3), (tel: +82-031-0817-1038)]
Student [ID: 21433015, Hwang, S-S, GPA:  2.72, (1989,  1, 10), (tel: +82-053-0817-1005)]
Student [ID: 21177017, Park, S-S , GPA:  4.12, (1989,  7, 10), (tel: +82-034-0817-1098)]
Student [ID: 21255340, Hong, G-M , GPA:  3.57, (1990,  2,  5), (tel: +82-055-0810-5678)]
Student [ID: 21766056, Zang, S-M , GPA:  4.45, (1990,  3, 15), (tel: +82-010-9112-1600)]
Student [ID: 21822075, Yoon, S-M , GPA:  4.37, (1990,  4,  5), (tel: +82-053-0811-1550)]
Student [ID: 21911000, Kim, G-M  , GPA:  3.57, (1990, 10,  5), (tel: +82-053-0805-1234)]
Student [ID: 21644054, Lee, K-M  , GPA:  3.35, (1991,  5, 15), (tel: +82-010-9112-9876)]
Student [ID: 21133053, Gong, G-W , GPA:  2.35, (1992,  6, 25), (tel: +82-055-0543-4321)]
Student [ID: 21588053, Choi, Y-H , GPA:  3.85, (1992,  9, 25), (tel: +82-053-0845-5764)]
Student [ID: 21021053, Song, W-B , GPA:  4.47, (1993,  5, 15), (tel: +82-034-0345-1234)]
Student [ID: 21010053, Kwak, S-B , GPA:  4.43, (1994, 11, 15), (tel: +82-002-0897-8778)]
```

```
/* Example of Handling an Array of Struct Student (4)  */

            case 8:
                sortStudents_by_TelNumber(students, num_students);
                printf("After sorting students by increasing
                    order of student's telephone number:\n");
                printStudents(students, num_students);
                printf("\n");
                break;
            default:
                break;
            } // end of switch

            shuffleStudents(students, num_students);
               // shuffle array for next experiments

        } // end of while
}
```

```
Input menu = 8
After sorting students by increasing order of student's telephone number:
Student [ID: 20911042, Bhang, S-H, GPA:  3.75, (1987, 12, 25), (tel: +82-002-0123-4567)]
Student [ID: 21010053, Kwak, S-B , GPA:  4.43, (1994, 11, 15), (tel: +82-002-0897-8778)]
Student [ID: 21766056, Zang, S-M , GPA:  4.45, (1990,  3, 15), (tel: +82-010-9112-1600)]
Student [ID: 21644054, Lee, K-M  , GPA:  3.35, (1991,  5, 15), (tel: +82-010-9112-9876)]
Student [ID: 21399019, Ahn, D-J  , GPA:  3.21, (1988, 10,  3), (tel: +82-031-0817-1038)]
Student [ID: 21021053, Song, W-B , GPA:  4.47, (1993,  5, 15), (tel: +82-034-0345-1234)]
Student [ID: 21177017, Park, S-S , GPA:  4.12, (1989,  7, 10), (tel: +82-034-0817-1098)]
Student [ID: 21911000, Kim, G-M  , GPA:  3.57, (1990, 10,  5), (tel: +82-053-0805-1234)]
Student [ID: 21822075, Yoon, S-M , GPA:  4.37, (1990,  4,  5), (tel: +82-053-0811-1550)]
Student [ID: 21433015, Hwang, S-S, GPA:  2.72, (1989,  1, 10), (tel: +82-053-0817-1005)]
Student [ID: 21588053, Choi, Y-H , GPA:  3.85, (1992,  9, 25), (tel: +82-053-0845-5764)]
Student [ID: 21133053, Gong, G-W , GPA:  2.35, (1992,  6, 25), (tel: +82-055-0543-4321)]
Student [ID: 21255340, Hong, G-M , GPA:  3.57, (1990,  2,  5), (tel: +82-055-0810-5678)]
```

memory allocation for struct variable

```
typedef struct
{
        char c2;
        char c1;
        int i1;
        int i2;
} Struct_ccii;

typedef struct
{
        char c1;
        int i1;
        char c2;
        int i2;
} Struct_cici;

typedef struct
{
        int i1;
        char c1;
        int i2;
        char c2;
} Struct_icic;
```

memory allocation for struct variable

```
/* main test_structs */
#include <stdio.h>
#include "Structs.h"

void main()
{
        Struct_ccii ccii;
        Struct_cici cici;
        Struct_icic icic;

        printf("Size of struct_ccii : %d\n", sizeof(ccii));
        printf("Size of struct_cici : %d\n", sizeof(cici));
        printf("Size of struct_icic : %d\n", sizeof(icic));

}
```

```
Size of struct_ccii : 12
Size of struct_cici : 16
Size of struct_icic : 16
```

memory allocation for struct variable

◆ Considerations

- memory alignment for high performance

Oral Test 9

실습 9 Oral Test

Q 9.1 구조체 배열을 동적으로 생성하는 절차를 예를 들어 설명하라.

Q 9.2 구조체를 포인터로 가리키게 한 후, 구조체의 각 항목을 포인터를 사용하여 읽거나 쓸 수 있게 하는 방법에 대하여 예를 들어 설명하라.

Q 9.3 비트 단위 구조체를 어떻게 선언하며, 비트 단위 구조체를 사용하면 어떤 장점이 있는가에 대하여 예를 들어 설명하라.

Q 9.4 4바이트 크기의 정수형 (integer) 데이터를 저장할 때 CPU 종류에 따라 Little Endian과 Big Endian에 따라 서로 다르게 저장되는 byte ordering (바이트 저장 순서)의 차이점에 대하여 예를 들어 설명하라.