객체지향형 프로그래밍과 자료구조 실습 1

1.1 큰 난수 배열의 생성

1) 32767 보다 큰 난수를 중복되지 않게 생성하여 주어진 배열에 담아주는 C/C++ **void genBigRandArray (int *pArray, int num_rands)**를 구현하라. 이 함수는 정수 배열의 주소 (pArray)와 배열의 크기(num_rands)를 인수로 전달 받으며, 0 ~ num_rands-1 범위의 중복되지 않는 난수를 생성하여 배열 pArray[]에 담아준다. num_rands의 값은 RAND_MAX인 32767보다 클 수 있으며, 최대 10,000,000까지의 값을 가질 수 있도록 구현하여야 한다.

1.2 구조체 struct Student 설계 및 구현

1) 학생들의 정보를 표현하기 위한 구조체 **struct Student**를 구현하라. 학생의 정보로는 5가지의 속성을 가진다: st_id (integer student ID), name (up to 15 characters), department (up to 15 characters), grade (1 ~ 8), GPA (double).
학생 구조체 struct Student는 별도로 분리된 헤더파일 "**Student.h**"에 저장하도록 구현하고, 학생 구조체에 관련된 함수들의 함수 원형을 포함하여야 한다.

2) 학생 구조체에 관련된 **void initStudents(Student students[], int student_id[], int num_students)** 함수를 C/C++ 프로그램으로 구현하라. 이 함수에는 구조체 Student의 배열 students[], 학번이 저장되어 있는 정수배열 student_id[]와 학생수 num-students가 인수로 전달된다. 학번이 포함된 정수배열 student_id[]에는 중복되지 않은 학번이 포함되어 있다. 구조체 struct Student에서 학번을 제외한 나머지 항목들을 rand() 함수를 사용하여 임의로 설정할 것. 이름 (name)은 5 ~ 15 문자를 포함하도록 하고, 처음 문자는 반드시 대문자 ('A' ~ 'Z')가 되어야 하며, 나머지는 소문자 ('a' ~ 'z')가 되도록 할 것. 학과는 3 ~ 5개의 대문자료 표시할 것 (예: ICE - Information and Communication Engineering). 학년을 나타내는 grade의 값은 1 ~ 8의 값을 가지게 하고, 종합 성적을 나타내는 gpa는 0.00 ~ 99.99 범위의 double 형 실수 값을 가지도록 할 것.

3) 학생 구조체에 관련된 **void printStudent(Student *pSt)** 함수를 C/C++ 프로그램으로 구현하라. 이 함수는 주어진 학생 구조체의 내용을 출력한다. 출력 포멧은 (st_id, last_name, first_name, department, grade, GPA)으로 할 것.

4) 학생 구조체에 관련된 **void printStudents(Student *students, int num_students)** 함수를 C/C++ 프로그램으로 구현하라. 이 함수는 학생들의 배열을 출력하며, printStudent() 함수를 사용한다.

5) 학생 구조체에 관련된 **void printStudentID(Student *students, int num_students)** 함수를 C/C++ 프로그램으로 구현하라. 이 함수는 학생의 정보 중에서 학번만을 출력한다.

6) 학생 구조체 정보 중 학번 만을 출력하며, 큰 배열의 첫 부분과 끝 부분만을 출력하는 **void printBigArrayOfStudent_IDs(Student *students, int size_array, int size_first_last_block)** 함수를 C/C++로 작성하라. 이 함수 호출에서 전달되는 인수 size_first_last_block 은 학생 배열의 첫 부분과 끝 부분에서 출력하는 학생 수를 지정하며, 한 줄에 10명씩 출력하도록 한다. 중간 부분은 ". . . ."을 출력하여 생략되었다는 표시를 하라.

7) 학생 구조체에 관련된 **void printStudentGPA_ID(Student *students, int num_students)** 함수를 C/C++ 프로그램으로 구현하라. 이 함수는 학생의 정보 중 성적과 학번을 (GPA, st_id) 형식으로 출력한다.

8) 학생 구조체 배열에서 성적과 학번을 출력하며, 큰 배열의 첫 부분과 끝 부분만을 출력하는 함수 **void printBigArrayOfStudent_GPA_IDs(Student *students, int size_array, int size_first_last_block)**를 C/C++로 작성하라. 이 함수 호출에서 전달되는 인수 size_first_last_block 은 학생 배열의 첫 부분과 끝 부분에서 출력하는 학생 수를 지정하며, 한 줄에 10명씩 성적과 학번을 출력하도록 한다. 중간 부분은 ". . . ."을 출력하여

생략되었다는 표시를 하라.

1.3 구조체 배열의 선택 정렬과 이진 탐색을 위한 함수 구현

1) 학생 구조체 배열을 선택 정렬하는 C/C++ 함수 void selectionSortStudentsBy_ID (Student students[], int num_students)를 구현하라. 이 함수는 학생들을 학번의 오름차순으로 정렬한다. 정렬된 결과는 printBigArrayOfStudent_IDs()함수를 사용하여 출력할 것.
2) 학생 구조체 배열을 선택 정렬하는 C/C++ 함수 **void selectionSortStudentsByGPA_ID (Student *students, int num_students)**를 구현하라. 이 함수는 학생들을 종합 평점(GPA)를 기준으로 내림차순 정렬하며, 만약 GPA가 동일할 경우 학번의 오름차순으로 정렬한다. 정렬된 결과는 **printBigArrayOfStudent_GPA_IDs()**함수를 사용하여 출력할 것.
3) 학생 구조체 배열을 탐색하는 C/C++ 함수 **Student * binarySearchStudentByID(Student *students, int num_students, int st_id)**를 구현하라. 이 함수는 학생들의 배열을 학번 기준으로 이진 탐색 기능을 수행하며, 탐색 결과는 **printStudent(Student *pSt)** 함수를 사용하여 출력한다.

1.4 main()

```
/* main.cpp */
#include <stdio.h>
#include <stdlib.h>
#include "Student.h"
#define NUM_STUDENTS 5000
#define NUM_SEARCH 5
#define NUM_FIRST_LAST_BLOCK 100

void genBigRandArray(int* pArray, int num_rands);

void main()
{
        int *student_ids;
        Student *students, *pSt;
        int student_search[NUM_SEARCH] = { 1, 123, 999, 2500, 4999 };

        student_ids = (int *)calloc(NUM_STUDENTS, sizeof(int) );
        students = (Student *)calloc(NUM_STUDENTS, sizeof(Student) );
        genBigRandArray(student_ids, NUM_STUDENTS);
        initStudents(students, student_ids, NUM_STUDENTS);

        //printStudentIDs(students, NUM_STUDENTS);
        printf("\n\n");
        printf("Initialized array of students : \n");
        printBigArrayOfStudent_IDs(students, NUM_STUDENTS, NUM_FIRST_LAST_BLOCK);

        printf("\n\n");
        printf("Selection sorting the array of students by (increasing order of ID) : \n");
        selectionSortStudentsBy_ID(students, NUM_STUDENTS);   // non-increasing order
        printBigArrayOfStudent_IDs(students, NUM_STUDENTS, NUM_FIRST_LAST_BLOCK);

        printf("\n\n");
        printf("Selection sorting the array of students by (decreasing order of GPA, increasing order of ID) :
        \n");
        selectionSortStudentsByGPA_ID(students, NUM_STUDENTS);   // non-increasing order
        printBigArrayOfStudent_GPA_IDs(students, NUM_STUDENTS, NUM_FIRST_LAST_BLOCK);

        printf("\n\n");
        printf("Selection sorting the array of students by (increasing order of ID) : \n");
        selectionSortStudentsBy_ID(students, NUM_STUDENTS);   // non-increasing order
        printBigArrayOfStudent_IDs(students, NUM_STUDENTS, NUM_FIRST_LAST_BLOCK);
```

```
		printf("\n\n");
		printf("Searching student with student_ID : \n");
		for (int i = 0; i < NUM_SEARCH; i++)
		{
			pSt = binarySearchStudentByID(students, NUM_STUDENTS, student_search[i]);
			printf("Student search by ID (%4d) : ", student_search[i]);
			if (pSt != NULL)
				printStudent(pSt);
			else
				printf("NULL - student was not found !!");
			printf("\n");
		}
}
```

1.5 실행 결과 (예시)

```
Initialized array of students :
1955 4012  516  462  280 4585 3635 2539 2102 1572 2892  889  638   44 4319 4022 2666 3304 2555 2774
3715 4629 3528 2999 1172 1740 1173  275    5 1050 3723 3762 2290 1662  400 4333  249 1510 2709 3372
 695 4466 3448 3264 2316 3431 1091 2954 1706 4013 3859  410 4602  642 3250 1873 4704 1189 4309    8
1538 1284 3407 2614 4860  950 4412 1318 4367 4713 2263  687 1669 4270 1042 4638  352 3791 4871 2239
3769 1347 2001 4686 1324 3252  749 2463  508 1542 3703 2962 2988 1003 3997 3147 2197 1697  898 4928
 . . . . .
2474  243 3209 3790 1578  572 3324 1745  288 3141  308 1827  213 4786 4772 1787  930  810 4864 2487
4909 4836 4463 3706 2058 4070 1848 3093 1192 2592 2336 4634 2680 1552 3146 2060 3204  221 2462 4525
4823 2181 4919 3484 3579 1601  519 2186 3491 1329 4006 1484 3206 1979 3034 1586 1064 2312  454 1883
 723 3770 1629 2384 2337 3768 1807 4644 2599 3203 2960 4979  227 4131 4708  685 1904  739 4145 3112
 164 1208 2446 2937 1886 1674 4347 4974 2326 2547 4182 4793 1455 2920  205 1729 2712 1642 3645 4965


Sorting array of students by non-decreasing order of ID :
   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
  20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37   38   39
  40   41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56   57   58   59
  60   61   62   63   64   65   66   67   68   69   70   71   72   73   74   75   76   77   78   79
  80   81   82   83   84   85   86   87   88   89   90   91   92   93   94   95   96   97   98   99
 . . . . .
4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919
4920 4921 4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939
4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 4958 4959
4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979
4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 4995 4996 4997 4998 4999


Sorting array of students by (decreasing order of GPA, increasing order of ID) :
(99.98, 4093) (99.95, 1142) (99.95, 4934) (99.93, 4597) (99.92, 4472) (99.89, 4387) (99.87, 4628) (99.85, 3988) (99.84, 2300) (99.83, 1362)
(99.83, 3668) (99.82, 4936) (99.79, 1513) (99.77, 4099) (99.70, 3977) (99.69, 1737) (99.68, 3700) (99.67,  576) (99.66, 2045) (99.60, 1969)
(99.59, 4550) (99.58, 1854) (99.56, 2922) (99.55, 3309) (99.53, 2670) (99.53, 3446) (99.48, 2369) (99.48, 2458) (99.47, 1300) (99.47,   15)
(99.46, 1992) (99.45, 1729) (99.43, 4744) (99.41, 4994) (99.35, 4467) (99.30, 1031) (99.30, 2581) (99.26, 3225) (99.25, 2598) (99.24, 1269)
(99.24, 3453) (99.18, 4638) (99.13, 3257) (99.12,   38) (99.12, 3077) (99.01, 2768) (99.00, 2118) (98.99,  714) (98.99, 3861) (98.97, 2320)
(98.91, 2645) (98.88, 2295) (98.86, 2181) (98.84, 4437) (98.79, 3985) (98.77, 4113) (98.72, 1358) (98.69, 3565) (98.68, 2180) (98.63,  517)
(98.60,  998) (98.58,  602) (98.57, 2682) (98.53,  977) (98.49, 3943) (98.43, 3299) (98.41, 4834) (98.37,  589) (98.36,  173) (98.36, 1140)
(98.36, 2447) (98.34, 1152) (98.31, 4793) (98.29, 2612) (98.28,  364) (98.24,  238) (98.22, 1701) (98.22, 4670) (98.20, 1991) (98.20, 2389)
(98.19,  895) (98.19, 4199) (98.17, 3057) (98.14, 1933) (98.10, 3056) (98.06,  113) (98.05,  248) (98.05, 3103) (98.05, 4576) (98.04,  118)
(98.02,  719) (98.01, 3314) (97.96,  219) (97.96, 1732) (97.96, 2238) (97.93, 4959) (97.92,  898) (97.90,   68) (97.90,  243) (97.90, 3272)
 . . . . .
( 1.97, 2844) ( 1.96, 4534) ( 1.93, 1911) ( 1.92, 4808) ( 1.87, 1116) ( 1.80,  395) ( 1.80, 1806) ( 1.79,  903) ( 1.78,  703) ( 1.76,  491)
( 1.72, 1840) ( 1.71,  421) ( 1.70, 4345) ( 1.70,  542) ( 1.69, 2235) ( 1.67, 2814) ( 1.67, 2335) ( 1.65, 4995) ( 1.59, 1578) ( 1.58, 1925)
( 1.56, 4494) ( 1.56, 3928) ( 1.55,  220) ( 1.55, 3745) ( 1.52, 4103) ( 1.51, 3148) ( 1.49, 4828) ( 1.45, 1342) ( 1.45, 3218) ( 1.45, 1968)
( 1.43, 4715) ( 1.43, 2055) ( 1.43,  882) ( 1.39, 1292) ( 1.39, 4218) ( 1.37, 2076) ( 1.36, 3051) ( 1.36,  543) ( 1.31,  347) ( 1.31, 2614)
( 1.29,  299) ( 1.27,  850) ( 1.25, 3753) ( 1.25, 1500) ( 1.24,  693) ( 1.22, 4699) ( 1.18, 4498) ( 1.15, 3909) ( 1.13, 3837) ( 1.11,  869)
( 1.08, 3332) ( 1.07, 3497) ( 1.06, 2089) ( 1.04, 4470) ( 1.00, 4589) ( 0.98, 2206) ( 0.96, 3724) ( 0.94, 4898) ( 0.93, 4045) ( 0.90,  193)
( 0.90, 1404) ( 0.90,  269) ( 0.87, 2185) ( 0.85,  356) ( 0.84, 1168) ( 0.82, 4189) ( 0.81, 1497) ( 0.77, 3164) ( 0.76, 1024) ( 0.75, 1885)
( 0.71, 1486) ( 0.68,  319) ( 0.68, 4581) ( 0.68, 3987) ( 0.67, 2556) ( 0.64, 1266) ( 0.60,  733) ( 0.55,  704) ( 0.54, 3597) ( 0.54, 1498)
( 0.52,   27) ( 0.51, 3193) ( 0.48, 4518) ( 0.46, 3360) ( 0.43, 3858) ( 0.40, 4667) ( 0.38,  618) ( 0.38, 3111) ( 0.38, 2742) ( 0.37,  375)
( 0.35, 4538) ( 0.33, 1329) ( 0.27, 4722) ( 0.21,  620) ( 0.11, 3656) ( 0.09, 2277) ( 0.08,  586) ( 0.07,  407) ( 0.06, 1115) ( 0.00, 2788)

Sorting array of students by non-decreasing order of ID :
   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
  20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37   38   39
  40   41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56   57   58   59
  60   61   62   63   64   65   66   67   68   69   70   71   72   73   74   75   76   77   78   79
  80   81   82   83   84   85   86   87   88   89   90   91   92   93   94   95   96   97   98   99
 . . . . .
4900 4901 4902 4903 4904 4905 4906 4907 4908 4909 4910 4911 4912 4913 4914 4915 4916 4917 4918 4919
4920 4921 4922 4923 4924 4925 4926 4927 4928 4929 4930 4931 4932 4933 4934 4935 4936 4937 4938 4939
4940 4941 4942 4943 4944 4945 4946 4947 4948 4949 4950 4951 4952 4953 4954 4955 4956 4957 4958 4959
4960 4961 4962 4963 4964 4965 4966 4967 4968 4969 4970 4971 4972 4973 4974 4975 4976 4977 4978 4979
4980 4981 4982 4983 4984 4985 4986 4987 4988 4989 4990 4991 4992 4993 4994 4995 4996 4997 4998 4999


Searching student with student_ID :
Student search by ID (   1) : Student(ID:    1, Name:        Zgypneuqw, Dept:  RNUCT, Grade: 2, GPA: 41.79)
Student search by ID ( 123) : Student(ID:  123, Name:        Kidurlqmj, Dept:  ECCOM, Grade: 4, GPA: 97.42)
Student search by ID ( 999) : Student(ID:  999, Name:        Qnlhfjxos, Dept:  AQWPT, Grade: 1, GPA: 31.79)
Student search by ID (2500) : Student(ID: 2500, Name:         Hhoefvfv, Dept:  YOSDC, Grade: 7, GPA: 17.40)
Student search by ID (4999) : Student(ID: 4999, Name:   Qxltpflydwmebc, Dept:   RJFQ, Grade: 8, GPA: 97.76)
계속하려면 아무 키나 누르십시오 . . .
```

2. 2021-2 객체지향형 프로그래밍과 자료구조 실습 1 Oral Test

학번		성명		점수	

(1) 구조체와 배열의 차이점에 대하여 설명하라.
(핵심포인트: 구성요소들의 자료형)

(2) genBigRandArray(int *randArray, int num_rands) 가 0 .. num_rands-1 범위의 중복되지 않는 난수들을 생성하고, 주어진 동적 배열 randArray[]에 저장하는 기능에 대하여 설명하라. (핵심포인트: RAND_MAX(32,767)보다 더 큰 난수의 생성, 중복여부 확인)

(3) 구조체 배열에 대한 선택 정렬의 내부 절차에 대하여 예를 들어 설명하라.
(핵심 포인트: 선택의 세부 절차에 대한 설명)

(4) Windows 운영체제 환경에서 C/C++ 프로그램 모듈이 실행하는 시간을 마이크로 초 (micro-second) 단위로 측정하는 방법에 대하여 예를 들어 설명하라.
(핵심 포인트: 사용되어야 하는 함수, 포함되어야 하는 라이브러리, 측정 방법)