CSED101. Programming & Problem solving

Fall, 2014

Programming Assignment #4

(70 points)

정연규 (azureheaven@postech.ac.kr)

- ***Due***: **2014. 11. 29 23:59**
- ***Development Environment: Windows Visual Studio 2010***

- ***제출물***
  - **C code files (mystring.h, mystring.c, assn4.c)**
    - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.
    - 제출시, 파일을 압축해서 제출하지 말 것.
  - **보고서 파일** (.doc(x) or .hwp) 예) assn4.doc(x) 또는 assn4.hwp
    - AssnReadMe.pdf 를 참조하여 작성할 것.
    - 프로그램 실행 화면을 캡쳐하여 보고서에 포함시키고 간단히 설명 할 것.
  - 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

- ***주의사항***
  - 각 문제에 해당하는 요구사항을 반드시 지킬 것.
  - 컴파일 & 실행이 안되면 무조건 0점 처리된다.
  - **하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)**
  - 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 'POSTECH 전자컴퓨터공학부 부정행위 정의'를 따른다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)
  - 이번 과제에서 추가 기능 구현에 대한 추가 점수가 없습니다.
  - 과제를 작성하는 데 있어서 전역변수를 선언하여 사용할 수 없습니다.
  - 모든 문제의 출력 형식은 아래의 예시들과 최대한 비슷하게 작성한다. 출력형식은 문제 정의에 부합하고 예시와 크게 벗어나지 않는 한 감점하지 않는다.

Problem 1: 문자열 처리 함수 (20점)

(문제)

문자열 라이브러리에 포함되어 있는 함수 중 일부를 직접 작성해 보자.

(설명)

제공된 assn4.zip의 압축을 풀면 **mystring.h**와 **mystring.c**가 주어진다.

mystring.h는 같은 문자열 처리 함수의 선언을 포함하고 있다. (그대로 사용하고 변경하지 말 것)

```
#ifndef MY_STRING_H
#define MY_STRING_H

int mystrlen(char *str);
char *mystrcpy(char *toStr, char *fromStr);
char *mystrcmp(char *str1, char *str2);
char *mystrcat(char *str1, char *str2);
char *mystrchr(char *str, char ch);
int myatoi(char *str);

#endif
```

위 함수의 구현부를 포함한 **mystring.c**를 작성하라. 각 함수의 정의는 다음과 같다.

(1) int mystrlen(char *str)

NULL 문자를 제외한 문자열의 길이를 반환한다. 빈 문자열의 경우 0을 반환한다.

예제)

```
printf("%d₩n", mystrlen("cs101")); //결과: 5
```

(2) char *mystrcpy(char *toStr, char *fromStr)

문자열 복사 함수로 NULL 문자를 포함한 문자열 fromStr를 문자열 toStr 에 복사한 후, 문자열 toStr 의 시작 주소를 반환한다.

예제)

```
char str[256];
printf("%s₩n", mystrcpy(str, "Good Day")); // 결과: Good Day
printf("%s₩n", mystrcpy(str, "Hello")); // 결과: Hello
```

(3) char *mystrcmp(char *str1, char *str2)

문자열 str1과 str2의 대소를 비교한다 (대소문자 구분). 비교기준은 아스키코드표의 값을 기준으로 한다.

각 문자열의 첫 번째 문자부터 비교를 시작한다. 만일 같다면 두 문자가 다를 때까지나 NULL 에 도달할 때까지 계속 비교를 수행한다.

- 비교 중 str1의 문자가 작을 경우 -1, 클 경우 1을 반환한다.
  문자열이 길이가 같고 모든 문자가 같을 경우, 0을 반환한다.
- 비교 중 하나의 문자열이 먼저 끝에 도달할 경우, 끝난 문자열을 작다고 판단한다.

예제)

```
printf("%d₩n", mystrcmp("csed101", "csed103")); // 결과: -1
printf("%d₩n", mystrcmp("csed", "Csed")); // 결과: 1
printf("%d₩n", mystrcmp("csed", "cse")); // 결과: 1
printf("%d₩n", mystrcmp("csed", "csed103")); // 결과: -1
```

(4) char *mystrcat(char *str1, char *str2)

문자열 연결함수로 str1 의 끝에 str2 를 이어 붙인다. 즉, 문자열 str1 뒤의 NULL 문자는 str2 의 첫 번째 문자로 덮어 씌워지고 str2 의 NULL 문자는 남는다. 문자열 연결 후, 문자열 str1 의 시작 주소를 반환한다.

예제)

```
char str[256] = "Hello";
mystrcat(str, ", World");
printf("%s₩n", str); // 결과: Hello, World
```

(5) char *mystrchr(char *str, char ch)

문자 ch가 문자열 str 에 있는지 검색하는 함수로, 문자열 str의 처음에서부터 검색하여 첫 번째로 문자 ch를 찾을 때까지 검색한다. 문자가 있으면 그 곳의 주소를 반환하고 그렇지 않은 경우 NULL 을 반환한다.

예제)

```
char str[256] = "Hello World";
char *p;
p = mystrchr(str, 'o');
printf("%s₩n", p); // 결과: o World
p = mystrchr(p+1, 'o');
printf("%s₩n", p); // 결과: orld
```

(6) int myatoi(char *str)

문자열을 정수로 바꿔준다. 단, 문자열 str 은 모두 숫자들로만 이루어졌다고 가정하며, int 의 표현 범위를 넘어서는 문자열에 대해서는 고려하지 않는다.

빈 문자열의 경우 0을 반환한다.

예제)

```
char str[256] = "1201";
printf("%d₩n", myatoi(str) ); // 결과: 1201
```

Problem 2: 성적 관리 프로그램 (50점)

(문제)

파일로부터 학생에 대한 정보와 학생이 수강하는 과목 및 성적이 저장된 파일을 읽어서 성적을 관리하는 프로그램을 작성한다.

(주의사항)

- 프로그램은 4개의 메뉴로 이루어져 있으며, 파일로부터 리스트를 생성하는 부분과 각 메뉴는 각각의 독립적인 함수로 작성 되어야 한다. 즉, 최소한 5개의 함수가 선언되어야 한다.
- Problem 1에서 작성한 mystring.h 를 include 하여 사용한다. (표준 헤더 파일 <string.h>를 include 하여 사용할 수 없다.)

(설명 및 요구사항)

- 특정 학생의 정보 및 과목 정보는 구조체의 변수를 이용하여 표현되며, 프로그램 내부에서 동적 할당된 구조체 포인터의 배열을 이용하여 관리한다.
- 즉, 프로그램 내부의 학생 성적 리스트와 과목 목록 리스트는 아래에 정의된 구조체 포인터 배열을 이용해서 관리한다. 연결 리스트(linked list)와 같은 방법을 사용하지 않는다.
- 최대 학생수(MAX_STUDENT_NUM)는 30명, 최대 과목수(MAX_SUBJECT_NUM)는 50과목이라고 가정하며 한 학생이 최대 수강할 수 있는 과목수(MAX_TAKING_SUBJECT_NUM)는 8과목으로 제한한다.

```
#define MAX_STUDENT_NUM 30
#define MAX_SUBJECT_NUM 50
#define MAX_TAKING_SUBJECT_NUM 8

typedef struct
{
	char sub_code[8];
	char sub_name[41];
	int credit;
	int nums;
}SUBJECT;

typedef struct
{
	char sub_code[8];
	char grade[3];
}TAKING_SUBJECTS;

typedef struct
{
	int id;
	char name[11];
	char dept[7];
	int level;
	int sub_num;
	TAKING_SUBJECTS subjects[MAX_TAKING_SUBJECT_NUM];
}STUDENT;
```

```
…

int main()
{
    STUDENT* stu_list[MAX_STUDENT_NUM];
    SUBJECT* sub_list[MAX_SUBJECT_NUM];
```

(1) 학생 성적 리스트 stu_list는 아래의 구조체 STUDENT* 형의 배열을 이용해서 관리한다.

- stu_list의 각 원소들은 학생 1명의 정보에 해당하는 STUDENT 구조체에 대한 주소를 가리키는 포인터이며, 각 학생에 대한 STUDENT 구조체는 학생이 추가될 때마다 각각 동적할당 받아 사용한다.

```
stu_list[i] = (STUDENT *)malloc(sizeof(STUDENT)); // i 는 학생수
```

- 학번(id)은 8자리의 정수(ex. 20150001)로 구성되며, 학번은 식별자(identifier)로 중복되지 않는다.
- 이름(name)은 공백 없는 한글 또는 영어 문자열(한글 최대 5글자, 영어 최대 10글자)
  ※ 한글 문자 1개는 2byte의 저장공간이 필요하다.
- 학과(dept)는 공백 없는 한글 문자열로 최대 3글자
- 학년(level)은 1 ~ 4 까지의 숫자
- sub_num은 현재 이수 중인 과목 수
- 수강과목(subjects)은 최대 8과목으로 구조체 TAKING_SUBJECTS 형의 배열로 구성
  - 수강과목코드(sub_code)는 알파벳과 숫자의 조합으로 7개의 문자열
  - 성적(grade)은 A+, A0, A-, … , F 형태로 구성된다.

(2) 과목 목록 리스트 sub_list는 구조체 SUBJECT* 형의 배열을 이용해서 관리한다.

- sub_list의 각 원소들은 한 과목의 정보에 해당하는 SUBJECT 구조체에 대한 주소를 가리키는 포인터이며, 각 과목에 대한 SUBJECT 구조체는 과목이 추가될 때마다 각각 동적할당 받아 사용한다.

```
sub_list[i] = (SUBJECT *)malloc(sizeof(SUBJECT));   // i 는 과목수
```

- 과목명(sub_name)은 공백 없는 한글 문자열로 최대 20자
- 학점(credit)은 한 자리 숫자
- 수강인원(nums)은 최대 3자리 숫자
- 과목코드명(sub_code)은 알파벳과 숫자의 조합으로 7개의 문자열
  - 코드명은 식별자로 중복되지 않는다.
  - 코드 목록은 알파벳 순서로 항상 정렬된 상태를 유지한다.
    과목코드는 네 자리의 문자와 세 자리의 숫자로 구성된다. 앞의 네 자리 문자는 교과목 개설학과, 뒤의 세 자리의 숫자는 임의의 고유번호를 나타낸다. 개설학과가 같은 경우에는 뒤의 세자리 숫자의 오름차순 정렬을 한다. 예를 들면, PHYS104, PHYS106 순서 이다

- 윈도우의 커맨드 라인에서 다음과 같이 입력하여 프로그램 실행은 다음과 같이 한다. 노란색으로 표시된 문자는 사용자 입력에 해당한다.

```
C:₩> assn4.exe students.txt
```

  - 파일명을 입력하지 않은 경우, default로 "students.txt"로부터 데이터를 읽는다.
  - 파일명이 입력된 경우, 입력된 파일로부터 데이터를 읽는다.
  - 파일명의 최대길이는 20자이며 파일명에는 공백이 없다고 가정한다.
- 아래의 모든 예시에 있는 대로 입력 시, 해당 예제와 동일하게 동작해야 한다. 출력 형식은 아래의 예시들과 최대한 비슷하게 작성한다. (출력형식은 문제 정의에 부합하고 예시와 크게 벗어나지 않는 한 감점하지 않는다.)
- 프로그램이 정상적으로 작동하기 위해 아래 경우에 예외 처리가 필요하다.

  (1) 파일 읽기에서 읽을 파일이 존재하지 않을 경우: 아래의 메시지 출력 후, 프로그램을 종료한다.

```
파일을 읽어올 수 없습니다.
계속하려면 아무 키나 누르십시오 . . .
```

  (2) 프로그램이 제시한 선택 목록에 없는 정수 범위의 입력이 들어온 경우

  예를 들면, 사용자가 1~4 사이의 선택을 해야 하나, 5나 6등의 입력을 넣은 경우를 말하며, **"올바른 입력이 아닙니다."** 라는 메시지 출력 후, 다시 입력을 받는다.
- 프로그램을 실행시키면, 텍스트 파일 "students.txt"로부터 학생의 정보를 읽는다. 파일의 구성과 내용은 아래와 같다.

```
20150001 홍길동 컴공 1
CSED101 프로그래밍과문제해결 3 A0
PHYS106 일반물리개론II 3 B+
PHYS104 일반물리실험II 1 A0
LIFE103 일반생명과학 3 A-
GEDU101 글쓰기 2 B0
MATH120 응용선형대수 3 A+
GEDU152 검도 1 A+

20150023 김기리 컴공 1
MATH120 응용선형대수 3 C0
CSED101 프로그래밍과문제해결 3 F
LIFE103 일반생명과학 3 B-
GEDU101 글쓰기 2 B+

20150135 김지민 화공 1
LIFE103 일반생명과학 3 A0
PHYS106 일반물리개론II 3 B-
PHYS104 일반물리실험II 1 A0
CSED101 프로그래밍과문제해결 3 A-
MATH120 응용선형대수 3 C0
```

- 각 학생에 대한 정보는 한 줄 띄움으로 구별한다.
- 입력파일 내의 학생의 정보는 이미 학번으로 정렬되어 있다고 가정하며, 따라서 학생 목록을 저장 및 출력할 필요가 있을 경우에는 파일에 입력된 순서대로 수행한다.
- 각 학생에 대한 정보 중 첫 번째 줄은 다음과 같은 형태로 주어진다. (각 데이터들은 하나의 공백으로 구분된다.)

  이름 학번 학과 학년
- 두 번째 줄부터는 그 학생이 수강하는 과목에 대한 정보로 아래의 형태로 주어지며, 최소한 1과목 이상을 수강한다.

  수강과목코드 수강과목명 학점 성적
- 한 학생이 수강하는 과목수는 최대 8과목으로 제한한다.

● 프로그램이 실행되면, 아래와 같은 메뉴 화면이 출력되고, 메뉴 입력을 기다린다.

```
**** 성적 관리 프로그램 ****
1. 학생별 성적
2. 과목별 성적
3. 저장하기
4. 종료
>>
```

1. 학생별 성적

- 숫자 1 을 입력 시, 출력 예시는 아래와 같다.

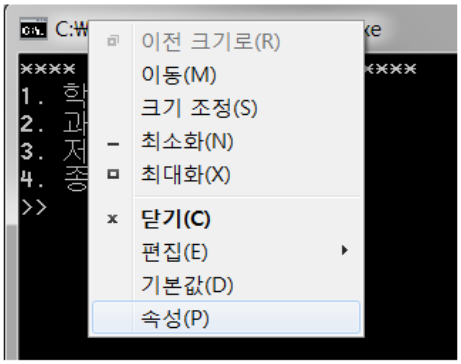
  각 학생 별 학번, 이름, 학과, 학년, 과목 코드 및 각 성적, 평점 평균과 비고를 출력한다.

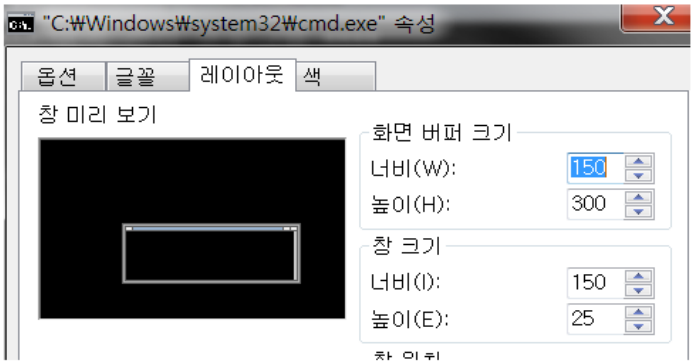
```
>> 1

학번       | 이름    | 학과(학년) | CSED101 | GEDU101 | GEDU152 | LIFE103 | MATH120 | PHYS104 | PHYS106 | 평점평균 | 비고
20150001 | 홍길동 | 컴공(1)    | A0       | B0       | A+       | A-       | A+       | A0       | B+       | 3.76     | 우등생
20150023 | 김기리 | 컴공(1)    | F        | B+       | -        | B-       | C0       | -        | -        | 1.88     | 학사경고
20150135 | 김지민 | 화공(1)    | A-       | -        | -        | A0       | C0       | A0       | B-       | 3.17     | -
```

- 과목코드 목록은 알파벳 순서로 정렬하여 출력한다.
- 학업성적에 따른 평점은 아래와 같다.

성적	A+	A0	A-	B+	B0	B-	C+	C0	C-	D+	D0	D-	F
평점	4.3	4.0	3.7	3.3	3.0	2.7	2.3	2.0	1.7	1.3	1.0	0.7	0.0

- 각 학생별 평균 평점은 각 과목에 대한 (평점x학점)을 하여 합한 후, 전체 이수 학점으로 나눈 값이다.
- 비고에 아래에 해당하는 학생을 표시한다.
  - 우등생: F 가 없는 자로 15학점 이상 이수하고 평점 평균이 3.6 이상인 자
  - 학사경고: 평점 평균이 2.0에 미달한 자
- 참고
  - 상태 표시창에 우클릭 -> 속성 -> 레이아웃으로 가면 실행창 크기를 조절 가능





2. 과목별 성적

숫자 2 를 입력 시, 출력 예시는 아래와 같다.

서브 메뉴로 과목 목록이 아래와 같이 출력된다. 0을 선택하면 상위 메뉴로 돌아간다. 해당 과목의 번호를 입력하면 해당 과목 정보와 과목을 듣는 학생들의 명단과 학점이 출력되고 마지막에 과목평균이 출력된다. 아래의 예제는 과목번호로 1을 선택했을 경우이다.

```
>> 2

과목 목록
  1 CSED101 프로그래밍과문제해결
  2 GEDU101 글쓰기
  3 GEDU152 검도
  4 LIFE103 일반생명과학
  5 MATH120 응용선형대수
  6 PHYS104 일반물리실험II
  7 PHYS106 일반물리개론II
나가기(0) >> 1

과목명: 프로그래밍과문제해결(CSED101), 학점: 3학점, 총 수강생수: 3

학번       | 이름       | 학과(학년)   | 학점
20150001 | 홍길동    | 컴공(1)      | A0(4.0)
20150023 | 김기리    | 컴공(1)      |  F(0.0)
20150135 | 김지민    | 화공(1)      | A-(3.7)

** 과목평균: 2.57

과목 목록
  1 CSED101 프로그래밍과문제해결
```

과목 정보로 과목명, 과목 코드, 학점, 총 수강생 수가 출력되어야 한다.
그리고, 그 아래에 각 수강 학생의 정보로 학번, 이름, 학과, 학년, 학점이 출력되어야 한다
실행 후, 과목 목록이 다시 출력되고, 0을 누른 경우 상위 메뉴로 돌아간다.

3. 저장하기

숫자 3 을 입력 시, 출력 예시는 아래와 같다.
학생별로 저장할지, 과목별로 저장할지 선택할 서브메뉴가 나오고 선택에 따라 저장할 파일명을 입력하게 된다. 입력하는 파일명에는 공백이 없으면서 최대길이는 20자라고 가정한다.

```
>> 3

목록
  1 학생별
  2 과목별
나가기(0) >> 1

학생별 성적을 저장할 파일명 입력: student_score.txt
학생별 성적 저장완료!!

목록
  1 학생별
  2 과목별
나가기(0) >> 2
과목별 성적을 저장할 파일명 입력: subject_score.txt
과목별 성적 저장완료!!

목록
```

(1) 학생별로 저장되는 성적 파일의 예시는 아래와 같다. 최상위 메뉴에서 1(학생별 성적)을 선택한 경우와 화면에 출력되는 내용을 파일에 저장하면 된다.

학번	이름	학과(학년)	CSED101	GEDU101	GEDU152	LIFE103	MATH120	PHYS104	PHYS106	평점평균	비고
20150001	홍길동	컴공(1)	A0	B0	A+	A-	A+	A0	B+	3.76	우등생
20150023	김기리	컴공(1)	F	B+	-	B-	C0	-	-	1.88	학사경고
20150135	김지민	화공(1)	A-	-	-	A0	C0	A0	B-	3.17	-

(2) 과목별로 저장되는 파일의 예시는 아래와 같다. 모든 과목에 대한 성적을 파일로 저장하면 된다.

```
No. 1
과목명: 프로그래밍과문제해결(CSED101), 학점: 3학점, 총 수강생수: 3

학번       | 이름       | 학과(학년)  | 학점
20150001 | 홍길동    | 컴공(1)     | A0(4.0)
20150023 | 김기리    | 컴공(1)     |  F(0.0)
20150135 | 김지민    | 화공(1)     | A-(3.7)

** 과목평균: 2.57

No. 2
과목명: 글쓰기(GEDU101), 학점: 2학점, 총 수강생수: 2

학번       | 이름       | 학과       | 학년  | 학점
20150001 | 홍길동    | 컴공       |    1  | B0(3.0)
20150023 | 김기리    | 컴공       |    1  | B+(3.3)

** 과목평균: 3.15

…

No. 7
과목명: 일반물리개론II(PHYS106), 학점: 3학점, 총 수강생수: 2

학번       | 이름       | 학과       | 학년  | 학점
20150001 | 홍길동    | 컴공       |    1  | B+(3.3)
20150135 | 김지민    | 화공       |    1  | B-(2.7)

** 과목평균: 3.00
```

4. 종료

최상위 메뉴에서 4를 선택하면, 프로그램을 종료한다.

프로그램을 종료하기 전, 동적할당을 받아 사용했던 모든 메모리를 반드시 할당해제 하도록 한다.