

과제#2

[총 15점: 6 문항]
(Q1~Q6: 각 문항 3점)

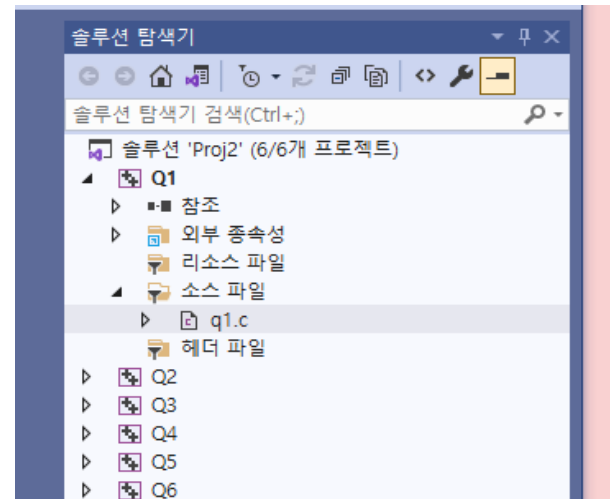
※ 15점 초과 점수는 절삭

컴프실 2022 (Spring)

cyclops@ajou.ac.kr

제출물

- 각 문항의 해답을 **소스코드 포함하여 하나의 문서 파일**로 (여러 개의 파일인 경우 하나의 압축 파일로) 제출합니다. 프로그램을 VS C로 구현한 경우, 아래 예와 같이, 각 문항의 프로그램이 개별 Project로 등록된 하나의 솔루션으로 구성하고, 이 솔루션 폴더를 압축한 파일을 제출하여도 됩니다..
- ※ 프로그램 작성시, 적절한 주석은 필수입니다.
- ※ 실행되지 않는 코드는 0점 처리됩니다.
- ※ 각 문항의 해답을 같은 (동일) 솔루션 내의 개별 Project로 추가하여 구현할 수 있습니다. 각 문항 별 해는 main 함수를 가진 프로그램으로 작성되어, 선택적으로 실행시킬 수 있습니다. 이 경우도, MS VS의 **솔루션 폴더를 압축**하여 제출하면 됩니다. (솔루션을 생성할 때, 어떤 폴더를 사용하는지 **확인하고, 제출 시 착오 없길 주의하세요.**)
- ※ '.sln' 파일은 솔루션 관련 정보를 가진, VS가 관리하는 파일입니다. 이 파일만 제출하는 실수를 하지 마세요.



Q1. 카드 게임 [3 점]

- 7장의 카드를 나누어 가지고, **아래 규칙에 따라 계산한** 카드의 숫자의 합이 가장 작은 사람이 이기는 게임입니다.

- ① 카드의 숫자가 같은 쌍의 카드는, 한 장의 하나 (1) 작은 숫자의 카드로 교환해 준다.
- ② 새로운 카드를 받은 후에도, 같은 숫자의 카드 쌍이 있으면, 계속해 위와 같은 교환을 반복한다. '1'의 카드가 쌍인 경우는 그냥 버린다.
- ③ 예, 받은 카드가 6 3 1 2 4 5 6 이면,
 $1\ 2\ 3\ 4\ 5\ 6\ 6 \rightarrow 1\ 2\ 3\ 4\ 5\ 5 \rightarrow 1\ 2\ 3\ 4\ 4 \rightarrow 1\ 2\ 3\ 3 \rightarrow 1\ 2\ 2 \rightarrow 1\ 1 \rightarrow 0$
- ④ 예, 받은 카드가 5 3 1 2 4 6 7 이면. $1 + 2 + 3 + 4 + 5 + 6 + 7 \rightarrow 28$
- ⑤ 카드의 숫자는 1에서 13까지 자연수로 제한한다.

- 이 게임에서 받은 7장 카드의 합을 구하는 C 프로그램을 작성합니다.

- ① 7장의 카드의 숫자는 난수 발생 함수, **rand**를 사용해 얻습니다.
(※ 난수 SEED 설정 안 함. → **srand** 함수 호출 안 함.)
- ② 받은 카드의 숫자들을 나열하고, 위의 규칙에 따른 카드 숫자의 합을 구해 출력합니다.
(입/출력 예시 참조)
- ③ 입력 문자가 '\n' (Enter)이면 위 과정을 반복 수행합니다.
- ④ 입력 문자가 '\n' (Enter)가 아니면 종료합니다.

Q1 <입/출력 예시>

*** Play Game ***

7 Cards given: 4 8 4 2 4 3 13

7 Cards sorted: 2 3 4 4 4 8 13

4 Cards left: 1 4 8 13 --> sum = 26

*** Play Game ***

7 Cards given: 11 8 1 8 7 5 10

7 Cards sorted: 1 5 7 8 8 10 11

5 Cards left: 1 5 6 10 11 --> sum = 33

*** Play Game ***

7 Cards given: 1 13 13 3 1 13 12

7 Cards sorted: 1 1 3 12 13 13 13

3 Cards left: 3 11 13 --> sum = 27

*** Play Game ***

7 Cards given: 1 13 6 7 7 3 12

7 Cards sorted: 1 3 6 7 7 12 13

5 Cards left: 1 3 5 12 13 --> sum = 34

x

Q2. 격자 경로 [3점]

- 행의 수가 N 이고 열의 수가 M 인 격자의 각 칸에 1부터 $N \times M$ 까지의 번호가 첫 행부터 시작하여 차례로 부여됩니다. 격자의 어떤 칸은 ○ 표시(Mark)가 되어 있습니다. (단, 1번 칸과 $N \times M$ 번 칸은 ○ 표시 대상이 아닙니다. 또한, ○ 표시가 되어 있는 칸은 최대 한 개입니다. 즉, ○ 표시가 된 칸이 없을 수도 있습니다.) 행의 수가 3이고 열의 수가 5인 격자에서 각 칸에 번호가 1부터 차례대로 부여된 예는 아래와 같고. 이 격자에서는 8번 칸에 ○ 표시가 되어 있습니다.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

- 격자의 1번 칸에서 출발한 어떤 로봇이 아래의 경로 이동 조건을 만족하면서 $N \times M$ 번 칸으로 이동합니다.
- 경로 이동 조건
 - 로봇은 한 번에 오른쪽에 인접한 칸 또는 아래에 인접한 칸으로만 이동할 수 있다. (즉, 대각선 방향으로서는 이동할 수 없다.)
 - 격자에 ○로 표시된 칸이 있는 경우엔 로봇은 그 칸을 반드시 지나가야 한다.

1	→	2	→	3		4		5
6		7		8	↓	9	→	10
11		12		13	↓	14	→	15

- 위에서 보인 것과 같은 격자가 주어질 때, 로봇이 이동할 수 있는 여러 경로 중 두 가지 예는 아래와 같다.
 - 1 → 2 → 3 → 8 → 9 → 10 → 15
 - 1 → 2 → 3 → 8 → 13 → 14 → 15
- 격자에 관한 정보, N, M, Mark가 주어질 때 로봇이 앞에서 설명한 두 조건을 만족하면서 이동할 수 있는 서로 다른 경로가 총 몇 개나 되는지 찾고, 각각의 경로를 출력하는 프로그램을 작성합니다. (입/출력 방법은 예시를 따릅니다.)
- 상기 입/출력 과정을 반복해 수행할 수 있어야 하고, N이 0일 경우 종료합니다. (N, M의 범위는 1에서 5까지로 제한합니다.)

Q2 <입/출력 예시>

Enter N, M and Mark (0<N,M<6): 3 5 8

Paths from 1 to 8:

1 2 3 8

1 2 7 8

1 6 7 8

Paths from 8 to 15:

8 9 10 15

8 9 14 15

8 13 14 15

of Paths stopping at 8: 9

Enter N, M and Mark (0<N,M<6): 3 5 16

Error: Out of Range

Enter N, M and Mark (0<N,M<6): 3 0 8

Error: Out of Range

Enter N, M and Mark (0<N,M<6): 6 3 8

Error: Out of Range

Enter N, M and Mark (0<N,M<6): 3 6 8

Error: Out of Range

Enter N, M and Mark (0<N,M<6): 0 0 0

Bye!!!

Enter N, M and Mark (0<N,M<6): 3 5 0

Paths from 1 to 15:

1 2 3 4 5 10 15

1 2 3 4 9 10 15

1 2 3 4 9 14 15

1 2 3 8 9 10 15

1 2 3 8 9 14 15

1 2 3 8 13 14 15

1 2 7 8 9 10 15

1 2 7 8 9 14 15

1 2 7 8 13 14 15

1 2 7 12 13 14 15

1 6 7 8 9 10 15

1 6 7 8 9 14 15

1 6 7 8 13 14 15

1 6 7 12 13 14 15

1 6 11 12 13 14 15

of Paths stopping at none: 15

※ Mark가 없는 경우: Mark = 0

Q3. 달팽이 배열 [3점]

- 회오리 (Spiral), 또는 달팽이 형태로 2차원 배열의 구성요소들의 값을 저장하거나 참조할 수 있습니다. 예로, 5x6의 배열인 경우, 아래와 같습니다.

오른쪽
회오리:

0	1	2	3	4	5
17	18	19	20	21	6
16	27	28	29	22	7
15	26	25	24	23	8
14	13	12	11	10	9

왼쪽
회오리:

0	17	16	15	14	13
1	18	27	26	25	12
2	19	28	29	24	11
3	20	21	22	23	10
4	5	6	7	8	9

- 크기, N 을 입력 받아, $N \times N$ (N 행, N 열) 배열의 구성 요소 값을 위의 예와 같이 왼쪽 회오리, 오른쪽 회오리 방식으로 채우로, 출력하는 C 프로그램을 작성합니다.
 - 프로그램 시작에 친절하게 “행렬 크기 N ($0 < N < 10$):”를 출력해 반복 입력할 수 있도록 하고, 숫자 ‘0’이 입력되면 프로그램을 종료합니다.
 - 입력 처리 후 결과 출력은 <입/출력 예시>를 따릅니다..

Q3 <입/출력 예시>

행렬 크기N [0<N<10]: 1

```
** Spiral Right & Left [1 x 1] **  
0      0
```

행렬 크기N [0<N<10]: 2

```
** Spiral Right & Left [2 x 2] **  
0  1    0  3  
3  2    1  2
```

행렬 크기N [0<N<10]: 5

```
** Spiral Right & Left [5 x 5] **  
0  1  2  3  4  0 15 14 13 12  
15 16 17 18  5  1 16 23 22 11  
14 23 24 19  6  2 17 24 21 10  
13 22 21 20  7  3 18 19 20  9  
12 11 10  9  8  4  5  6  7  8
```

행렬 크기N [0<N<10]: 8

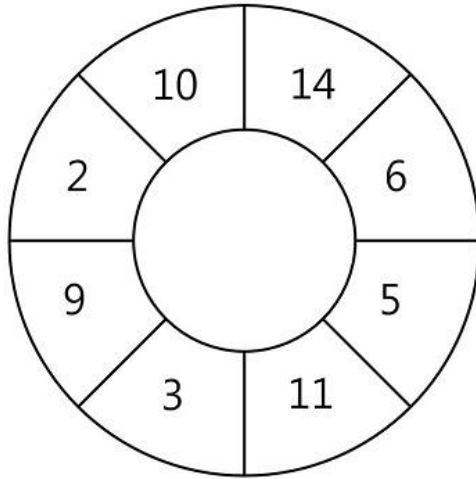
```
** Spiral Right & Left [8 x 8] **  
0  1  2  3  4  5  6  7  0 27 26 25 24 23 22 21  
27 28 29 30 31 32 33  8  1 28 47 46 45 44 43 20  
26 47 48 49 50 51 34  9  2 29 48 59 58 57 42 19  
25 46 59 60 61 52 35 10  3 30 49 60 63 56 41 18  
24 45 58 63 62 53 36 11  4 31 50 61 62 55 40 17  
23 44 57 56 55 54 37 12  5 32 51 52 53 54 39 16  
22 43 42 41 40 39 38 13  6 33 34 35 36 37 38 15  
21 20 19 18 17 16 15 14  7  8  9 10 11 12 13 14
```

행렬 크기N [0<N<10]: 0

Bye!!!

Q4. 스티커 뜯기 [3점]

- N개의 스티커가 원형으로 연결되어 있습니다. 다음 그림은 N = 8인 경우의 예시입니다.



- 원형으로 연결된 스티커에서 몇 장의 스티커를 뜯어내어 뜯어낸 스티커에 적힌 숫자의 합이 최대가 되도록 하고자 합니다. 단 스티커 한 장을 뜯어내면 양쪽으로 인접해있는 스티커는 찢어져서 사용할 수 없게 됩니다.
- 예를 들어 위 그림에서 14가 적힌 스티커를 뜯으면 인접해있는 10, 6이 적힌 스티커는 사용할 수 없습니다. 스티커에 적힌 숫자가 배열 형태로 주어질 때, 스티커를 뜯어내어 얻을 수 있는 숫자의 합의 최댓값을 계산 하는 c 프로그램을 작성합니다

- 스티커의 개수 N을 입력 받고, N 개의 스티커, 각 칸의 숫자를 난수발생 함수-rand()를 통해 생성하여, 위의 방법으로 계산한 최대 합을 출력합니다. 그리고, 뜯어낸 스티커에 적힌 숫자들도 같이 출력하도록 합니다. (반복하여 실행하고, N = 0 인 경우 종료합니다.)
 - ① sticker는 원형으로 연결된 스티커의 각 칸에 적힌 숫자가 순서대로 들어있는 배열로, 길이(N)는 1 이상 10 이하입니다.
 - ② sticker의 각 원소는 스티커의 각 칸에 적힌 숫자이며, 각 칸에 적힌 숫자는 1 이상 100 이하의 자연수로 제한합니다. (난수발생 함수 사용을 위한 SEED 설정은 하지 않습니다.)
 - ③ 원형의 스티커 모양을 위해 sticker 배열의 첫 번째 원소와 마지막 원소가 서로 연결되어 있다고 간주합니다.

Q4 <입/출력 예시>

```
# of Stickers: 10
Stickers: 42 68 35 1 70 25 79 59 63 65
* The BestSum = 289
* Stickers picked: 42[0] 35[2] 70[4] 79[6] 63[8]
# of Stickers: 8
Stickers: 6 46 82 28 62 92 96 43
* The BestSum = 246
* Stickers picked: 6[0] 82[2] 62[4] 96[6]
# of Stickers: 6
Stickers: 28 37 92 5 3 54
* The BestSum = 146
* Stickers picked: 92[2] 54[5]
# of Stickers: 5
Stickers: 93 83 22 17 19
* The BestSum = 115
* Stickers picked: 93[0] 22[2]
# of Stickers: 0
Bye!!!
```

※ 참고: 입/출력 예

```
# of Stickers: 6
Stickers: 28 37 92 5 3 54
* The Max Sum (123, 146) ==> 146
* Stickers picked: 92[2] 54[5]
```

※ 첫 스티커를 뜯지 않았을 때의 합

※ 첫 스티커를 뜯었을 때의 합

참고 용

Q5. Rail Fence Cipher [3점]

- 문자열을 규칙에 따라 재배치하여 평문을 암호화할 수 있습니다. 이러한 암호화/복호화 알고리즘을 구현하여, 주어진 암호문은 평문으로, 평문은 암호문으로 출력합니다.

[illegible]

- **5행X25열** 표에 회색배경이 칠해진 칸에 문장의 문자를 재배치하여 암호문을 만듭니다. 즉, 주어진 평문, "**DREAMS COME TRUE.**"를 위 같이 문자를 배치하여, 문자들을, 첫 행에서부터 아래 행으로, 왼쪽에서부터 오른쪽으로, 차례로 모아 내려 가면, 암호문, "**DO.RCMEE@EUAS@RMT**"를 만들 수 있습니다.

※ 편의상 문장에 들어 있는 띄어쓰기 빈칸(space)을 ㉠로 표시함.

- 위의 방법으로, 주어진 암호문을 복호화하고, 평문을 암호화할 수 있는 C 프로그램을 작성하고, 아래 암호문은 복호화하고, 평문은 암호화하여 출력합니다.

- 암호문(1): CNⓈOIGYDNTⓈRONCEETEV
- 암호문(2): 길도있어서길이다끝에이람나긔되사는는

- 평 문(1) : THEORY**⑥**OF**⑥**RELATIVITY
- 평 문(2) : 철학적통찰물리학적직관수학적기교의놀라운결합

Q5 <암호화 출력 처리 예시 >

평 문 : DREAMS⑥COME⑥TRUE.

암호문: DO.RCMEE⑥EUAS⑥RMT

평 문 : 꿈의무게를느끼고견디어야꿈을이룬다

암호문: 꿈견다의고디룬무끼어이게느야을를꿈

Q5 <복호화 출력 처리 예시>

암호문: DO.RCMEE⑥EUAS⑥RMT

평 문 : DREAMS⑥COME⑥TRUE.

암호문: 꿈견다의고디룬무끼어이게느야을를꿈

평 문 : 꿈의무게를느끼고견디어야꿈을이룬다

※ ⑥는 띄어쓰기 빈칸으로 출력하면 됨.

※ 참고사항: Wide Character Set을 사용하여 한글/영어 같이 사용할 수 있도록 구현합니다. (교재 예제 12-22 수정 본 참조)

```
/* 교재 12장 예제 12-22 수정 본 */
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <locale.h>
```

```
int main() {
```

```
    char str[] = "ABC언어";
```

```
    wchar_t uni_str[] = L"ABC언어";
```

```
_wsetlocale(LC_ALL, L"korean"); //지역 설정: 대한민국 (사용문자 -> 한글)
```

```
    printf("str: %s\n", str);
```

```
    printf("sizeof(str) is %d\n", sizeof(str));
```

```
    printf("strlen(str) is %d\n", strlen(str));
```

```
    printf("uni_str: %ls\n", uni_str);
```

```
    printf("sizeof(uni_str) is %d\n", sizeof(uni_str));
```

```
    printf("wcslen(uni_str) is %d\n", wcslen(uni_str));
```

```
    return 0;
```

```
}
```

Q6. 며칠째? [3 점]

- 태어난 날로부터 몇 일을 살았는지 계산하는 프로그램을 작성합니다. [\(5장: 실습 #11\)](#)
 - 생년월일의 년 (year), 월 (month), 일 (day)을 각각 정수로 입력 받아, 생년월일과 요일, 그리고, 오늘이 태어나서 **몇 번째 날**인지 출력합니다.
 - 위 (1) 항을 반복하여 실행하며, 생년월일의 년도가 1900 미만일 경우 종료합니다.
 - 종료 시 **오늘 날짜와 요일**, 그리고, 오늘이 **올해 몇 번째 날**인지 출력합니다.
- 오늘의 년(tyear), 월(tmonth), 일(tday)은 아래 함수를 호출하여 얻습니다.
→ `getToday(&tyear, &tmonth, &tday);`

```
#include <time.h>
int getToday(int *y, int *m, int *d) { // 오늘 년월일
    time_t current;
    struct tm *t;
    time(&current);
    t = localtime(&current);
    *y = t->tm_year + 1900;
    *m = t->tm_mon + 1;
    *d = t->tm_mday;
    return t->tm_yday; //올해 지난 날 수
}
```

- 1900년 1월 1일이 월요일**입니다. 이 날을 기준으로 요일을 계산할 수 있습니다.
- 입/출력 예 (다음 쪽)

Q6 <입/출력 예>

```
*** Days of Life ***
Enter the Day 1 (as yyyy.mm.dd) : 2022.1.1
The Day 1 of Life : 2022.01.01 (Saturday)
The number of days passed since the Day 1: 153
--> The Day 154 of Life
Enter the Day 1 (as yyyy.mm.dd) : 1953.3.12
The Day 1 of Life : 1953.03.12 (Thursday)
The number of days passed since the Day 1: 25285
--> The Day 25286 of Life
Enter the Day 1 (as yyyy.mm.dd) : 0.1.1
* Out of Bound: Year [0] < 1900
Today is 2022.06.03 (Friday),
The Day 154 of this Year.
```

※ Q5 입력 코드 예 → `getDayOne(&year, &month, &day);`

```
void getDayOne(int *y, int *m, int *d) { // 시작 년월일
    printf("Enter the Day 1 (as yyyy/mm/dd) : ");
    scanf("%d/%d/%d", y, m, d);
}
```

※ Q5 유용한 자료 예

```
int month[][12] = { { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }, // index 0: 평년
                    { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 } }; // index 1: 윤년
char* week[] = { "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" };
```