CSED101. Programming & Problem solving

Fall, 2015

Programming Assignment #2

(60 points)

이재국(ljk8918@postech.ac.kr)

- ***Due***: **2015.10.15 23:59**
- ***Development Environment***: **Windows Visual Studio 2010**

- ***제출물***
  - **C Code file (.c)**
    - 확장자를 포함한 소스파일 이름은 "**assn2.c**"로 할 것.
    - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 **주석**을 붙일 것.
  - **보고서 파일** (.doc(x) or .hwp) 예) assn2.doc(x) 또는 assn2.hwp
    - AssnReadMe.pdf 를 참조하여 작성할 것.
    - **프로그램 실행 화면을 캡쳐하여 보고서에 포함시키고 간단히 설명할 것.**
    - **명예서약(Honor code)**: 표지에 다음의 내용을 포함한다. "나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다." 보고서 표지에 명예서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.
  - 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

- ***주의사항***
  - 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 숙제에서 추가 기능 구현에 대한 추가 점수는 없습니다.

■ Problem: Blackjack 게임

(목적)

이번 과제를 통하여 조건문, 반복문, 사용자 정의 함수 및 라이브러리 함수 사용법을 익힌다.

(주의사항)

- 이번 과제는 함수를 정의하고 사용하는 방법을 익히는 문제이므로 사용자 정의 함수를 사용하지 않고 main함수에 모든 기능을 구현한 경우 감점 처리 함. (반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인한 후, 구현하도록 한다.)
- 프로그램 구현 시, main() 함수를 호출하여 사용하지 않는다.
- 전역 변수는 사용할 수 없으며, 아직 수업시간에 다루지 않은 배열 및 포인터는 사용하지 않는다.
- **문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.**

(설명)

간단한 Blackjack 게임을 구현 해 보자. 게임은 컴퓨터와 1:1로 이루어지며 컴퓨터가 딜러(Dealer)가 되고 사용자가 플레이어(Player)가 된다. Player와 Dealer는 1~11 사이의 카드를 랜덤하게 받게 되며, 각자 받은 카드의 합이 21점 또는 21점에 가까운 사람이 이기는 게임이다.
게임의 방법은 다음과 같다.
플레이어는 카드를 받기 전에 걸고 싶은 액수의 돈을 건다. 딜러는 카드를 플레이어 → 딜러 → 플레이어 → 딜러 순으로 나누어준다. 두 장씩의 카드를 받은 후 모두 공개 한다. 처음 2장의 카드가 21점이 된 것을 '블랙잭'이라고 하며, 21점의 카드를 가진 자가 게임에서 승리하게 된다.
플레이어와 딜러가 블랙잭이 아닌 경우, 먼저 플레이어는 자신의 카드의 합이 21점에 가까워지도록 하기 위해서, 추가로 카드를 받을지 여부를 결정하게 된다. 카드는 1장씩 몇 장이라도 요구할 수 있으며, 카드를 더 받지 않는 것이 유리하다고 판단하면 추가하지 않아도 된다. (플레이어가 21점을 넘은 경우, 게임에서 지며 베팅한 금액을 잃게 된다.)
플레이어의 카드 추가가 끝나면 딜러가 카드를 추가할 것인지 여부를 결정하게 되는데, 가진 카드의 총합이 16점이하이면 카드를 추가하고 17점이상이면 추가하지 않는다.
딜러의 점수와 비교해서 동점이면 무승부, 딜러보다 높으면 이기고, 낮으면 지게 된다.

I. 초기 선택 메뉴

프로그램을 실행하면, 아래처럼 사용자가 선택할 수 있는 메뉴가 출력되고, 사용자로 하여금 3가지 선택사항 중 한가지를 입력 받을 준비를 한다.

```
[ 블랙잭 게임 ]
================================
 1. 게임 설명
 2. 게임 시작
 3. 종료
================================
 선택:
```

1. 사용자가 1을 입력하면, 아래처럼 게임 설명을 출력한 후, 다시 초기 선택 메뉴를 출력하고 사용자 입력을 받을 준비를 한다.

```
========================== 게임 설명 ==========================
블랙잭은 카드의 합이 21에 가깝거나 21일 때 승리하는 게임이다.
플레이어와 딜러는 1~11사이의 카드를 랜덤하게 뽑을 수 있다. 카드를 받기 전 플레이어는 원하는 금액을 걸 수 있다.
딜러가 처음에 카드를 플레이어-딜러 순으로 한 장씩 총 두 장씩을 분배한다. 이 때 두 장의 카드의 총합이 21인 선수가 있는 경우 '블랙잭'으로 승리하게 된다.
딜러와 플레이어가 블랙잭이 아닌 경우 플레이어는 오픈된 카드를 보고 카드를 더 받을지(Hit) 말지(Stand) 결정한다. 카드의 합이 21이 넘어가면(Bust) 패배하게 되므로 21를 넘지 않는 가까운 수를 만드는 것이 관건이다.
플레이어가 카드를 모두 뽑은 후, 딜러가 카드를 더 뽑을지 결정한다. 딜러는 총 합이 17을 넘지 않으면 카드를 계속해서 더 뽑아야한다.
결과를 확인하여, 카드 합이 21이거나 21을 넘지 않는 큰 수를 가진 사람이 이기게 된다.
플레이어가 이길 경우 배팅한 금액의 두 배를 받고, 지면 잃는다. 같은 수가 나오면 무승부로 하여 베팅금을 돌려받는다.
==============================================================

[ 블랙잭 게임 ]
================================
 1. 게임 설명
 2. 게임 시작
 3. 종료
================================
선택:
```

2. 정수 1, 2, 3을 제외한 다른 값이 입력될 때는 다시 입력 받는다. 숫자 이외의 입력은 없다고 가정한다. (숫자에 대한 예외처리만 하면 됨)
3. 사용자가 3을 입력하면, 프로그램을 종료한다.

```
[ 블랙잭 게임 ]
================================
 1. 게임 설명
 2. 게임 시작
 3. 종료
================================
선택: 4
다시 선택하세요

[ 블랙잭 게임 ]
================================
 1. 게임 설명
 2. 게임 시작
 3. 종료
================================
선택: 3
계속하려면 아무 키나 누르십시오...
```

4. 이 부분을 메인 함수에서 반드시 스위치(switch)문을 사용하여 구현한다.
5. **<u>이 문제를 해결하기 위해 반드시 아래의 사용자 정의 함수를 정의하고 사용</u>**해야 하며, 아래 함수 외의 필요한 함수를 정의해서 사용할 수 있다.
   ➔ **<u>print_tutorial</u>**: 위의 tutorial을 출력하는 함수

II. 게임 시작 (2를 선택 시)

<u>아래 설명에서 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인한 후, 구현하도록 한다.</u> 설명에서 구현하라고 한 함수 외에 필요한 함수를 정의하고 사용할 수 있다.

1. 베팅 하기

- 사용자가 2를 선택하면, 아래의 그림처럼 초기 게임머니(소지금)로 10,000원이 지급되며 사용자(플레이어)로부터 게임에 베팅할 금액을 입력 받을 준비를 한다. (연속적인 게임을 위해서 게임 횟수와 소지금을 함께 표시한다.)
- 플레이어는 소지금내에서 베팅금을 입력한다.
  - 베팅금은 양수만 입력된다고 가정하며, 소지금이 0원이면 게임이 종료된다. (소지금 0원 이하로 게임이 종료되는 실행예제는 게임종료의 예제를 참고할 것.)
  - 소지금보다 많은 베팅금을 입력할 시 다시 입력할 것을 요청을 한다.
- 소지금내에서 베팅금을 입력하게 되면, 아래 예제처럼 현재 소지금이 베팅한 금액만큼 차감되어 출력되고, 베팅금액도 출력되며, 실제 게임이 시작된다.

```
선택: 2

[ 베팅 하기 ]
============================================================
=====  [ 블랙잭 게임  1 ]  [ 소지금: 10000 ]
============================================================
베팅 금액: 1000000
소지금보다 큽니다.
베팅 금액: 1000

============================================================
=====  [ 블랙잭 게임  1 ]  [ 소지금:  9000 ]  [ 베팅 금액: 1000 ]  =====
```

- **사용자 정의 함수 (반드시 정의하고 사용할 것)**
  ➔ **<u>check_betting_money</u>**: 플레이어에게 베팅 금액을 입력 받고, 입력 받은 금액이 소지금을 초과하지 않을 때, 플레이어가 입력한 베팅금을 반환

2. 카드 분배

- 실제 게임이 시작되면, Player와 Dealer는 1~11 사이의 카드를 랜덤하게 받게 되며, 카드는 플레이어 → 딜러 → 플레이어 → 딜러 순으로 각각 두 장을 받는다. 받은 카드는 플레이어와 딜러 둘 다 공개한다. (실제 게임과는 다르게 채점을 위해 딜러의 카드도 모두 공개한다.)

- 플레이어와 딜러 각각의 카드 총합을 아래와 같은 형태로 출력한다.
  (Player) 플레이어의 카드 총합 < =========> (Dealer) 딜러의 카드 총합

```
=====================================================
=====  [ 블랙잭 게임  1 ]  [ 소지금:  9000 ]  [ 베팅 금액: 1000 ]  =====
=====================================================
Player의 1번째 카드는  5입니다.       #
                                      #      Dealer의 1번째 카드는  8입니다.
Player의 2번째 카드는  1입니다.       #
                                      #      Dealer의 2번째 카드는  4입니다.
                  -------------------------------------------------
                    (Player)  6  <=========>  (Dealer) 12
                  -------------------------------------------------

                  [ Player's Turn ]
                  Hit or Stand? (h/s):
```

- 이 순서에서 카드의 합이 21점이 된 참가자(플레이어, 딜러)가 있는 경우 Blackjack을 출력하고 게임이 종료된다. 예제는 마지막 페이지를 참조하시오.
- 이 카드 분배 순서에서 참가자 각각의 카드의 합이 22점이 될 수 없도록 구현한다. 1번째 받은 카드가 11인데, 2번째 카드도 11을 받게 되었다면, 2번째 카드를 다시 받도록 하여 22점이 되지 않도록 한다. 즉, 카드 분배 순서에서 21점을 넘게 되어 게임이 끝나는 경우가 없도록 구현한다. (이 부분은 화면에 따로 출력하지 않음)
- 각각 받은 두 장의 카드의 합이 21점 보다 작은 경우, 플레이어 순서를 수행한다.
- **사용자 정의 함수 (반드시 정의하고 사용할 것)**
  - ➔ **<u>card_shuffle:</u>** 1~11 사이의 임의의 카드 숫자를 반환한다.
    - 이 함수를 이용하여 카드를 나누어줄 때, 함수 호출 후 0.3초가 지난 후에 카드를 주도록 한다.
    - Sleep 함수 이용(Sleep(300), 단위는 밀리초이다. 헤더는 <Windows.h>이다.)
    - 매 실행마다 다른 카드의 숫자가 생성되어 분배되도록 작성한다.

3. 플레이어 순서

- 이 순서에서 플레이어는 카드를 더 받을지(Hit), 받지 않을 지(Stand)를 결정할 수 있다. 카드를 추가로 받는 경우에는 h를 입력, 받지 않을 경우에는 s를 입력하며, 그 외의 입력은 고려하지 않는다.

```
                  [ Player's Turn ]
                  Hit or Stand? (h/s): h
                  -------------------------------------------------
                    (Player) 10  <=========>  (Dealer) 12
                  -------------------------------------------------
                  Hit or Stand? (h/s): h
                  -------------------------------------------------
                    (Player) 19  <=========>  (Dealer) 12
                  -------------------------------------------------
                  Hit or Stand? (h/s):
```

- 'h' (Hit)을 입력하여 카드를 추가로 받았을 때,
  - 카드의 합이 21점이 넘게 되면, 버스트(Bust)가 되어 딜러의 승리로 게임이 종료된다.
  - 카드의 합이 정확히 21점이 되면, 자동으로 딜러의 순서로 넘어간다.
- 's' (Stand)를 입력한 경우, 딜러 순서로 넘어간다.
- **사용자 정의 함수 (반드시 정의하고 사용할 것)**
  - ➔ **player_turn**: 카드를 추가로 받을 것인지를 물은 후 플레이어 카드 합을 반환. (함수 card_shuffle 을 이용하여 카드를 받는다.)

4. 딜러 순서

- 이 순서에서 딜러는 자신의 총합을 알고 있고 총합이 16점이하이면 카드를 추가하고 17점이상이면 추가하지 않는다.

```
Hit or Stand? (h/s): s

[ Dealer's Turn ]
--------------------------------------------------
  (Player)  19  <==========>  (Dealer) 16
--------------------------------------------------
--------------------------------------------------
  (Player)  19  <==========>  (Dealer) 22
--------------------------------------------------

[ Result ]
```

- 아래 실행예제는 카드 분배시에 딜러의 카드가 17점 이상인 경우로 카드 추가 없이 승패 판정으로 넘어간다.

```
Hit or Stand? (h/s): s

[ Dealer's Turn ]

[ Result ]
```

- **사용자 정의 함수 (반드시 정의하고 사용할 것)**
  - ➔ **dealer_turn**: 추가로 받을 것인지 위 기준으로 판별 후 딜러 카드 합을 반환. (함수 card_shuffle 을 이용하여 카드를 받는다.)

5. 승패 판정

- 딜러 순서가 끝나면 카드의 숫자의 총합을 비교하여 승패를 판정한다.
- 승패의 조건은 아래와 같다.
  - 처음 카드 분배 (2장씩)에서 블랙잭(10+11)인 경우, 해당 플레이어가 승리한다.
  - 카드의 합이 같은 경우 무승부이다.
  - 카드의 합이 21점이 넘는 참가자(플레이어 또는 딜러)가 패배한다. (게임 규칙에 따라 플레이어 순서에서 버스트가 되면 플레이어 패배, 딜러 순서에서 버스트가 되면 딜러가 패배한다.

- 카드의 합이 21점인 참가자가 승리한다. (블랙잭과 카드 추가로 달성한 21점은 다르고, 블랙잭이 우위이므로 구별한다.)
- 참가자들의 각각 카드 합이 21점을 넘지 않는 경우에는 21점에 더 가까운 참가자가 승리한다.

• 획득 금액
- 플레이어가 승리한 경우, 베팅한 금액의 2배를 받는다.
- 플레이어가 패배한 경우, 베팅한 금액을 잃게 된다.
- 무승부인 경우, 베팅한 금액을 돌려 받는다.

```
                ------------------------------------------------
                  (Player)  19  <=========>  (Dealer) 22
                ------------------------------------------------

                [ Result ]
                ------------------------------------------------
                Dealer Busts.
                Player Wins!
                베팅 금액 1000원의 두 배(2000원)을 획득하셨습니다.
                현재 소지금은 11000원입니다.
                ------------------------------------------------
                게임을 계속하겠습니까? (y/n): y

[ 베팅 하기 ]
=====================================================
=====  [ 블랙잭 게임  2 ]  [ 소지금: 11000 ]
=====================================================
베팅 금액: 2000

=====================================================
=====  [ 블랙잭 게임  2 ]  [ 소지금:  9000 ]  [ 베팅 금액: 2000 ]  =====
=====================================================
Player의 1번째 카드는  3입니다.       #
                                      #       Dealer의 1번째 카드는 5입니다.
```

• **사용자 정의 함수 (반드시 정의하고 사용할 것)**

➔ **judgement**: 플레이어와 딜러의 카드 합을 비교하여 승부를 판단하여 획득 금액을 결과값으로 반환 (승리시 배팅금의 두배, 패배시 0, 무승부시 배팅금만큼)

6. 게임 종료

- 게임의 승패가 결정된 후에, "게임을 계속하겠습니까? (y/n): "라는 메시지가 출력되고 사용자 입력을 기다린다.
- 이때, 플레이어가 y를 입력하면, 위의 실행예제처럼 이전 소지금을 가지고 게임을 이어나간다. n을 입력하면, 아래의 실행예제처럼 초기메뉴로 돌아간다. (y, n 외의 입력은 고려하지 않는다.)

```
                게임을 계속하겠습니까? (y/n): n

 [ 블랙잭 게임 ]
================================
 1. 게임 설명
 2. 게임 시작
 3. 종료
================================
선택:
```

- 게임 결과에서 소지금이 0원이되면 다음 실행예제처럼 게임을 계속할 것인지 묻지 않고 게임이 종료된 후, 초기 메뉴가 출력된다.

```
                ------------------------------------------------
                  (Player) 12   <=========>   (Dealer) 21
                ------------------------------------------------
                [ Result ]
                ------------------------------------------------
                Dealer Wins!
                베팅 금액 2000원을 잃었습니다.
                현재 소지금은 0원입니다.
                ------------------------------------------------
                소지금이 부족하여 게임을 이어나갈 수 없습니다.

 [ 블랙잭 게임 ]
================================
 1. 게임 설명
 2. 게임 시작
 3. 종료
================================
선택:
```

7. 실행예제

(1) 플레이어가 블랙잭인 경우

```
=====================================================
=====  [ 블랙잭 게임  1 ]  [ 소지금:  9000 ]  [ 베팅 금액: 1000 ]  =====
=====================================================
Player의 1번째 카드는 10입니다.       #
                                      #      Dealer의 1번째 카드는 7입니다.
Player의 2번째 카드는 11입니다.       #
                                      #      Dealer의 2번째 카드는 3입니다.
                  -------------------------------------------------
                    (Player) 21   <=========>   (Dealer) 10
                  -------------------------------------------------

                  [ Result ]
                  -------------------------------------------------
                  ★★★★★★★ Blackjack ★★★★★★★
                  Player Wins!
                  베팅 금액 1000의 두 배 (2000)를 획득하셨습니다.
                  현재 소지금은 11000원입니다.
                  -------------------------------------------------
                  게임을 계속 하시겠습니까? (y/n):
```

(2) 딜러가 블랙잭인 경우

```
=====================================================
=====  [ 블랙잭 게임  1 ]  [ 소지금:  9000 ]  [ 베팅 금액: 1000 ]  =====
=====================================================
Player의 1번째 카드는  4입니다.       #
                                      #      Dealer의 1번째 카드는 11입니다.
Player의 2번째 카드는  9입니다.       #
                                      #      Dealer의 2번째 카드는 10입니다.
                  -------------------------------------------------
                    (Player) 13   <=========>   (Dealer) 21
                  -------------------------------------------------

                  [ Result ]
                  -------------------------------------------------
                  ★★★★★★★ Blackjack ★★★★★★★
                  Dealer Wins!
                  베팅 금액 1000원을 잃었습니다..
                  현재 소지금은 9000원입니다.
                  -------------------------------------------------
                  게임을 계속 하시겠습니까? (y/n):
```

(3) 플레이어와 딜러가 동시에 블랙잭인 경우

```
=====================================================
=====  [ 블랙잭 게임  1 ]  [ 소지금:  9000 ]  [ 베팅 금액: 1000 ]  =====
=====================================================
Player의 1번째 카드는 10입니다.       #
                                      #       Dealer의 1번째 카드는 11입니다.
Player의 2번째 카드는 11입니다.       #
                                      #       Dealer의 2번째 카드는 10입니다.
                  ------------------------------------------------
                    (Player) 21 <=========>   (Dealer) 21
                  ------------------------------------------------

                  [ Result ]
                  ------------------------------------------------
                  ★★★★★★★ Blackjack ★★★★★★★
                  Draw
                  베팅 금액 1000원을 획득하셨습니다.
                  현재 소지금은 10000원입니다.
                  ------------------------------------------------
                  게임을 계속 하시겠습니까? (y/n):
```