**Term Project Report**



**수업명: 프로그래밍**

**교수님: 이남규 교수님**

**소속: 경영경제대학**

**학과: 응용통계학과**

**학번: 20122776**

**이름: 양명철**

**Term Project**

날짜 : 2018년 11월 9(금)

제출마감: 2018년 12월 9일(일)

제출방법: eclass 에서 제출(하나의 파일로 압축해서 제출)

* 프로그램 도큐먼트(.docx 또는 hwp 파일)
  + Cover page
  + 포커 게임 방법, 게임 규칙, 승자 판단 등과 같은 프로그램 설명
  + 포커 게임을 위한 자료구조, 함수 구성에 대한 설명
  + 프로그램 실행 결과 등
* C 프로그램 : 프로그램 작성 규칙에 맞게 작성(특히 comment)
  + 3장 강의자료참고, /\* \*/, // comment
  + 1개 이상의 사용자 정의 함수(user defined function)를 정의하여 사용하시오
  + 유효한 입력 값만 처리하도록(예외 처리) 프로그램을 해야 한다.

2단계 이상의 메뉴 레벨과 이번 학기에서 배운 내용을 활용하고, 각자 프로젝트 주제를 선택하여 개별 프로젝트를 수행 하시오.

* **주제 : 세븐 포커 카드 게임(seven poker card game)**
* 최대 7명에게 7장의 카드를 나누어 주는 포커게임을 만드시오.
* 게임 방법 : 아래의 Step을 게임을 종료할 때까지 반복한다.

Step 1: 사용자 자신을 제외한 게임의 참석자 수를 입력한다. (1 ~ 6)

Step 2: 52장의 카드를 랜덤 하게 섞는다.

Step 3: 참석자들에게 카드 1장씩 순서대로 3장을 나누어 준다.

Step 4: 각자 3장중에 1장씩 오픈한다.

* + - * 오픈하는 카드 선택은 3장의 카드 중 세번째 카드를 오픈한다.

Step 5: 상대방의 카드보다 자신의 카드가 좋지 않다고 판단되면 언제든 경기를 포기하는 다이(Die)를 하거나 베팅할 수 있다.

Step 6: 각자 일정 금액을 베팅한다.

Step 7: 참석자들에게 카드 1장씩 순서대로 나누어 준다.

Step 8: 카드를 한 장씩 나누어준 후 일정금액을 베팅을 반복 수행하는 방식으로 최대 7장까지 나누어 주도록 한다.

Step 9: 최종 7장까지 카드가 배분되고, 베팅이 끝났으면 승자를 판단하여 결과를 화면에 보인다.

Step 10: 이 과정을 반복한다.

* 자신이 차별성으로 내세울 수 있는 기능이 무엇인지 기술하여 부각시키시오.
* 모든 컴퓨터 플레이어는 이길 수 있는 높은 족보의 확률에 의거하여 판단하면서 게임을 한다.
* (확률 부분은 먼저 확정 패부터 확인한다. 그 다음에 각각 원페어일 경우 트리플과 포카드가 될 수 있는지 현재 뿌려진 카드를 제외한 카드 덱에서 해당 숫자가 있는지 체크 후 확률을 계산한다. 스트레이트 같은 경우는 4개까지 연결되고 하나가 모자랄 경우 앞뒤의 숫자를 앞 에와 같이 체크해서 확률 계산하고 투페어 같은 경우 투페어가 됬었을 때 각각 페어들의 숫자를 덱에 물어봐서 풀하우스 체크한다. 게임의 재미를 위해서 매턴마다 1/40의 확률로 죽거나 뻥카(페이크)배팅을 한다. 플레이어 모두 5장이상 받았을 때부터 해당 플레이어들은 다른 플레이어들의 오픈된 카드들만으로 분석해서 자신의 덱보다 높을 경우, 무조건 죽고 높지 않아도 후에 높아질 확률이있으면 다이를 한다. 다이하는 확률은 레벨별로 스트레이트가 나올 위험도 레벨 1 2 3 4, 플러쉬가 나올 위험도 레벨 1 2 3 4, 각각 족보에서 한 단계 위가 나올 위험도 레벨 1 2 3 4로 매긴 뒤, 1레벨 은 10% 2레벨은 25%, 3레벨은 33%, 4레벨은 50% 확률로 다이를 한다.)
* 카드를 화면에 나타낼 때 일반 문자열 보다는 카드 그림이 포함되도록 한다.

(상자 그리기, 카드 번호, 카드 모양,를 사용하여 좌표를 이용하여 세미 그래픽을 사용하여 표현하였다.)

* 카드를 섞는 과정을 동적으로 표현한다.

(동적 이펙트는 출력으로 카드 좌표 변경과 출력 순서를 활용하여 섞이는 것처럼 보이게 하였다. 실제로 섞는 부분은 총 100번의 반복문을 사용하여 0~ 51의 랜덤 상수값을 2개를 받아서 두 개의 위치를 섞는 방식으로 하였다.)

**<포커 게임 방법>**

Step 1: 처음에 사용자 자신을 제외한 게임의 참석자 수를 입력한다. (1~6)

Step 2: 52장의 카드를 랜덤 하게 섞는다.

Step 3: 참석자들에게 카드 1장씩 순서대로 3장을 나누어 준다.

Step 4: 각자 3장중에 1장씩 오픈한다.

– 오픈하는 카드 선택은 3장의 카드 중 세번째 카드를 오픈한다.

Step 5: 상대방의 카드보다 자신의 카드가 좋지 않다고 판단되면 언제든 경기를 포기하는 다이(Die)를 하거나 베팅할 수 있다.

Step 6: 각자 일정 금액을 베팅한다.

Step 7: 참석자들에게 카드 1장씩 순서대로 나누어 준다.

Step 8: 카드를 한 장씩 나누어준 후 일정금액을 베팅을 반복 수행하는 방식으로 최대 7장까지 나누어 주도록 한다.

Step 9: 최종 7장까지 카드가 배분되고, 베팅이 끝났으면 승자를 판단하여 결과를 화면에 보인다.

Step 10: 이 과정을 반복한다.

**<게임 규칙, 승자 판단, 프로그램 설명 >**

아무런 족보가 없다면 1번(Ace)가 제일 높으며, 그 다음은 K,Q,J 10,9… 순이다. 이 순서는 같은 족보를 가졌을 때도 같은 기준으로 높은 순서의 카드를 가진 사람이 이긴다. 숫자 이상의 족보로는 다음과 같은 순서로 족보들의 순위가 매겨진다.

원페어: 같은 숫자 카드 두 장 (43.8%)

투페어: 같은 숫자 카드 두 장이 두 벌 (23.5%)

트리플: 같은 숫자 카드 세 장 (4.83%)

스트레이트: 이어지는 번호로 5장 가장 높은 스트레이트는 마운틴(0.45%)이라 부르며 10부터 1까지의 카드 (스트레이트 플러시: 숫자가 연속되고 무늬가 같은 카드 5장 (0.0247%), 백 스트레이트: A, 2, 3, 4, 5 (.045%))

플러쉬: 숫자 상관 없이 같은 무늬 5장 (3.03%)

(무늬도 같은데 5장의 숫자가 이어져 있다면 스트레이트 플러쉬(0.0247%), 백 스트레이트 플러시: 무늬가 같은 A,2,3,4,5(0.0032%), 이어지는 숫자가 10부터 1이라면 로얄 스트레이트 플러쉬 포커에서 가장 높은 등급의 족보(0.0247%))

풀하우스: 같은 3장과 2장 (2.60%)

포커: 같은 번호 4장 (0.168%)

같은 족보일 경우,

만약 같은 족보라면 우선적으로 높은 번호의 숫자로 만들어진 족보가 이긴다. 포커에서는 1(Ace)가 높은 숫자이다. 그런데 숫자마저 같다면 모양(무늬)로 순서를 정한다. 포커의 모양 순서는 스페이드, 하트, 클로버, 다이아의 순으로 높은 등급이 포커의 규칙이다.

**< 문제 해결 방안 및 포커 게임을 위한 자료구조, 함수 구성에 대한 설명 >**

게임 시작후,

CardManagerSet() 함수로 전체 카드생성

GameSet() 으로 전체 플레이정보 초기화

게임 플레이 시작 시SetGame() 함수로 플레이어 7명 셋팅 밑 참가 인원제한

CardReSet() 전체 카드덱 초기화 한후

CardShuffle()함수로 카드 섞은 후

PushCard(3, StartIndex); 함수로 처음 카드 세장 드로우

첫플레이어가 자신의 카드 족보를

GradeCheck() 함수를 호출하여 받은 후

EnemyCardSort() 함수로 상대방 카드덱들의 오픈된카드 정보 수집

모인 정보를 가지고 판단하여 배팅

7장받을때까지 반복하여

최종승리자 체크

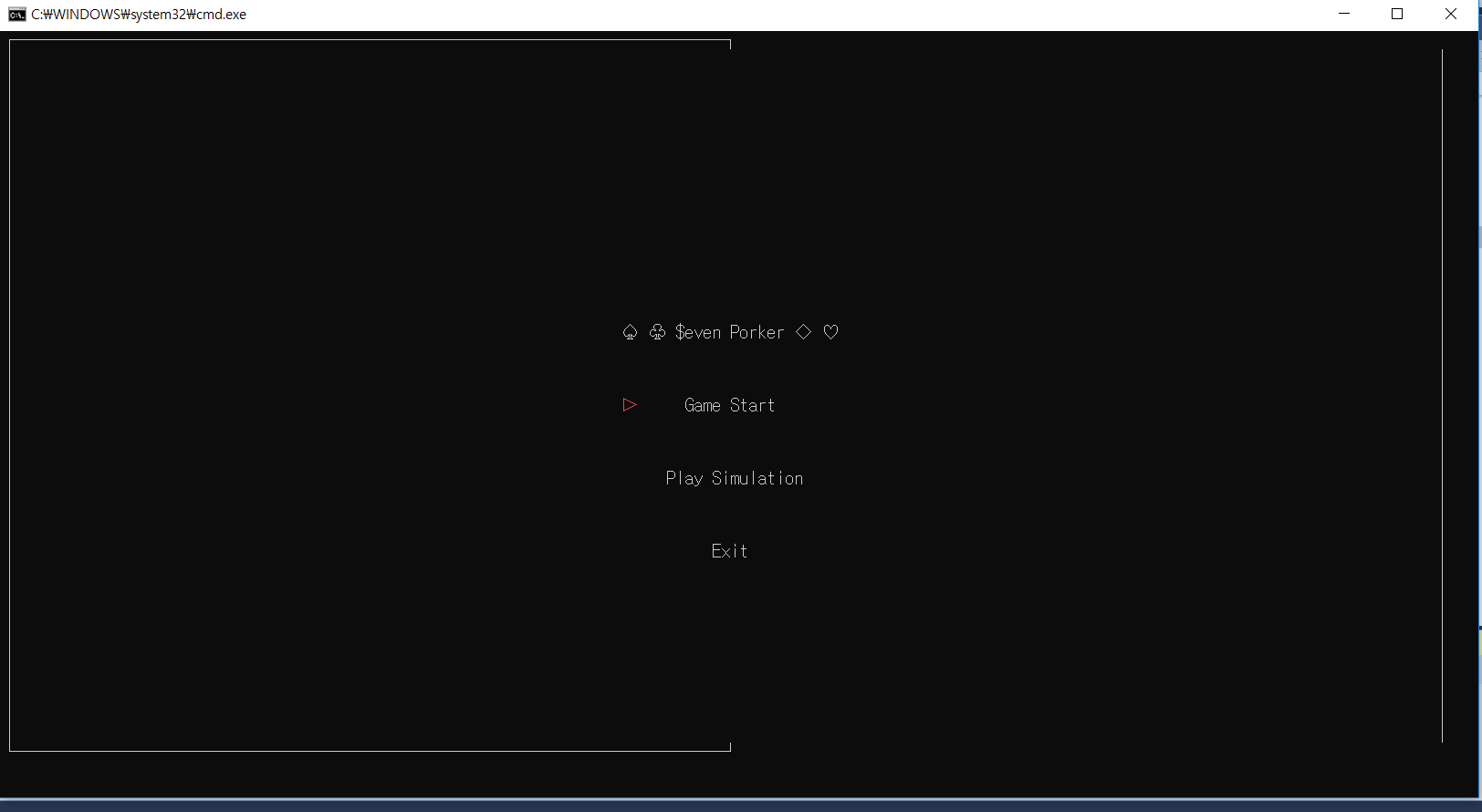
PlayerReset() 함수를 사용하여 잔액 0원인 플레이어 강제퇴장 시킨 후 남은 게임진행

혹시 잔액0원이 6명이면 한명 최종승리로 마무리

-다양한 함수를 만들어 모듈화 한다.

Gotoxy 함수는 좌표 이동을 한다. ShapeSetCard는 카드 모양을 설정 한다. CardManagerSet은 카드를 생성한다. ResetCard는 카드를 초기화한다. CardReSet은 카드 전체 초기화한다. DrawNum은 카드 번호를 그린다. DrawShape는 카드 모양을 그린다. CardDraw는 카드를 그린다. CardShuffle은 카드 섞기 및 모션을 드로우한다. GetCard는 랜덤으로 카드 보낸다. FindCardNum 부분은 보유 중인 카드 중 플레이어에게 안간 카드와 갔지만 가려져 있는 카드 중 동일한 숫자가 있는지를 체크한다. FindCardShape 부분은 보유 중인 카드 중 플레이어에게 안간 카드와 갔지만 가려져 있는 카드 중 동일한 모양이 있는지를 확인한다. SwapCardGrade함수와 SwapCheckCardInfo를 사용하여 섞는다. Reset은 카드체크 관련 중 변수를 리셋한다. CopyCard는 체크용 카드로 변환을 한다. 카드 오름차순 정렬(선택정렬)을 한다. PairCheck함수는 원페어 트리플 포카드 관련된 것을 체크한다. 같은 카드가 있는지 탐색한다. 등급이 현재보다 크면 갱신하고 등급이 같아도 숫자가 크면 갱신한다. 등급과 숫자가 같아도 모양이 크면 갱신한다. 체크한 등급이 원페어보다 크거나 같으면 다음 패 나올 확률을 체크한다. 페어 확률을 체크하고, 트리플일 경우 풀하우스를 체크한다. TwoPairCheck는 투페어나 트리플을 체크한다. 처음 나온 페어인지 체크한다. 그리고 두번쨰 나온 페어를 체크한다. 이전 페어와 숫자가 다른지를 체크한다. 트리플이 한 번이라도 나오면 풀하우스이다. 아니면 투페어이다. 현재 등급보다 찾은 등급이 높으면 갱신한다. 그리고 투페어가 풀하우스 될 확률을 체크한다. StraightCheck함수는 스트레이트를 체크한다. 연속숫자 개수를 체크하고 같은 숫자는 건너뛴다. 다음 확인 인덱스로 이동하여 최대 연속숫자인지 확인 후 갱신한다. Count가 5보다 크면 스트레이트로 확정을한다. 그리고 마운틴을 체크한다. Count가 4이상인 경우 스트레이트가 아닌 상태에서 백스트레이트를 체크한다. 플러쉬 유무를 체크하고 스트레이트 플러쉬, 로얄스트레이트플러쉬를 확인한다. 그리고 플러쉬 유무를 체크한다. 검사 경과 스트레이트가 현재 등급보다 높을 경우 갱신한다. 스트레이트 완성은 아니지만 4개 이상일 경우 스트레이트가 될 확률을 체크한다. FlushCheck는 플러쉬 확률을 체크한다. 같은 모양이 있으면 tmp 증가한다. 같은 모양이 3개 이상이며 이전 최대 같은 개수보다 클 경우 갱신한다. 그리고 플러쉬를 확정짓는다. 같은 모양이 5개는 안되지만 3개 이상일 경우 같은 모양의 남은 카드를 보고 확률을 본다. CardOpen함수는 카드를 연다. EraseCard함수는 카드 화면에서 지운다. DrawMidText함수는 중앙정렬 문자열을 출력한다. GetvoidString은 문자열만큼의 빈 공간 문자열을 반환한다. DrawPoint는 특수 문자를 그린다. ErasePoint는 특수 문자를 지운다. MenuSelectCursor는 메뉴선택 일반화 함수이다. DrawTitle은 처음 타이틀을 그린다. SmallBoxDraw는 작 작은 상자 그리기 좌표와 상자크기 필요하다 State = 플레이어상자 죽었는지 체크를 하여 false면 다이를 한다. BoxDraw는 상자를 그린다. SetPlayer는 사용자들을 설명하는 함수이다. ResetPlayerCard는 카드를 리셋한다. AllOpen은 전체카드를 연다. BetGold는 플레이어 배팅 돈관리를 한다. LastCardOpen은 마지막 카드를 연다. SetCard는 카드를 받는다. NameDraw함수는 이름을 그린다. EraseName함수는 이름을 지운다. MainBoxDraw는 자신의 카드 보관함을 그린다. PanWindCheck함수는 플레이 도중 나머지 전체가 죽었는디 확인을 한다. PlayerDelete는 플레이어 전체를 삭제한다. PlayerReset함수는 플레이어를 리셋한다. 그리고 Count가 1인 경우 최후의 1인으로 승리를 체크한다. WinDraw함수는 어떤 것으로 승리하였는지를 표시한다. SortGrade함수는 체크용 배열로 내림차순 정렬을 한다. 0이 가장 등급 높은 카드 족보이다. EnemyCardSort는 자신을 제외한 나머지 플레이어들의 등급을 확인한다. 가려진 카드는 제외된다. BetGoldDraw함수는 배팅 내용을 출력한다. GoldDraw함수는 돈을 그린다. PushCard함수는 카드 전체를 뿌린다. CardCount는 뿌릴 카드 장수이다. 1장인 경우 바로 오픈을 한다. Startindex는 처음 받을 플레이어 인덱스이다. SetGame은 게임 시작전 전체 초기화이다. GamePlay는 게임시작을 한다. Auto를 true로 하면 자동이고 false로 하면 수동이다. 플레이어 전체 카드를 리셋하고 박스를 그린다. 처음 3장을 그리고 남은 4장을 1장씩 그린다. 자신의 등급을 확인하고 게임 도중 전체가 다 죽으면 승리한다. 적들 카드 등급을 확인하고 턴이 플레이어가 아니거나 오토모드일 경우 자동으로 배팅을 한다. 확률적으로 무조건 죽거나 확률적으로 무조건 배팅을 한다. 카드를 총 5장 이상 받았을 시 자신의 등급보다 높은 카드를 확인하고 높은 카드가 있으면 바로 죽는다. 없으나 높아질 레벨이 있으면 확률에 따라 죽는다. Level에 따라 50%에서 25%까지이다. 그리고 판돈 올리고 판을 키운다. 게임 종료 후 살아있는 플레이어 모두 패 오픈을 한다. 그리고 승리를 표시한다.

**<프로그램 실행 결과 캡쳐>**



Game Start를 Enter를 누른다.

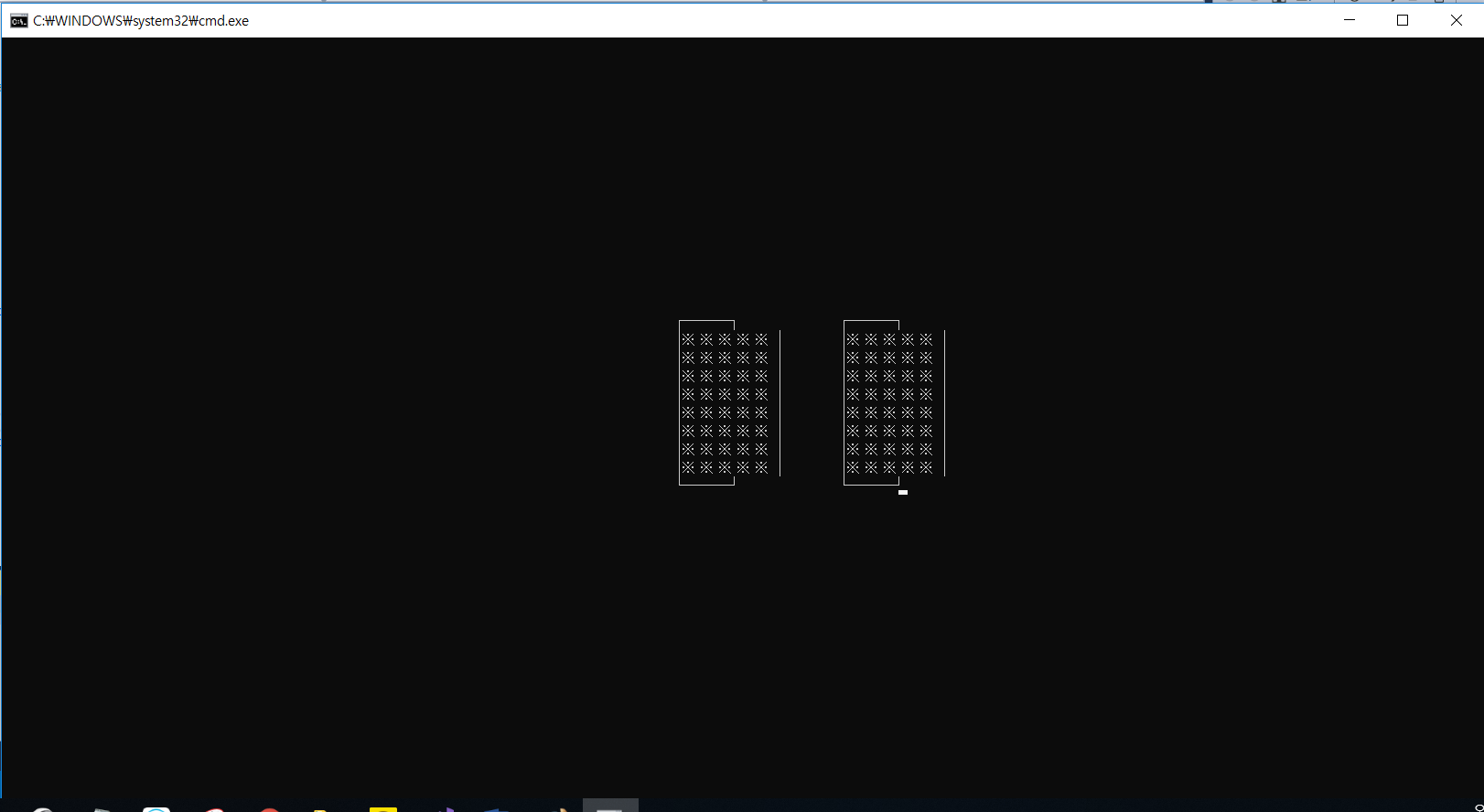


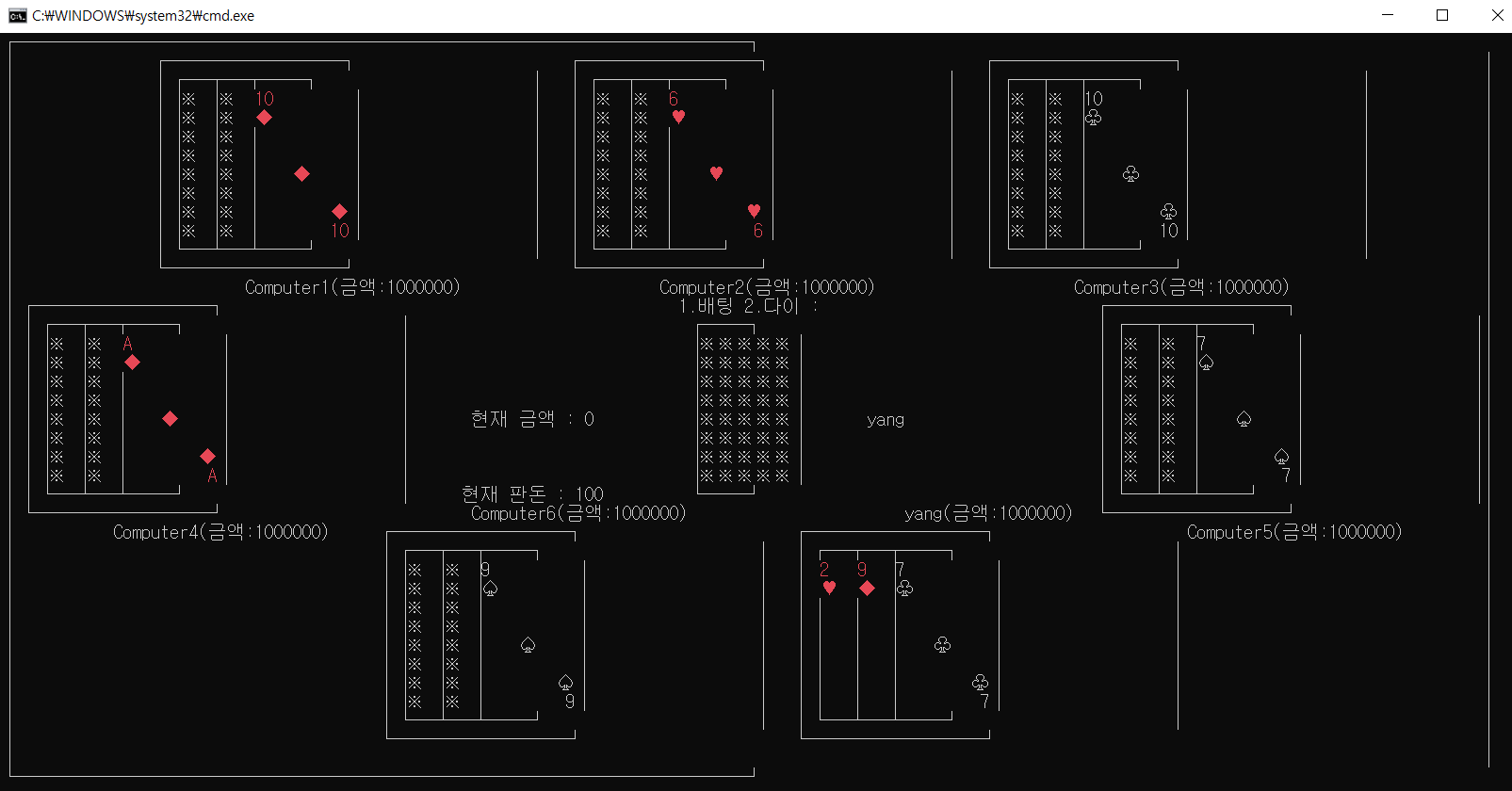
Player 이름을 입력 받아 작성한다.



Step 1: 사용자 자신을 제외한 게임의 참석자 수를 입력한다. (1 ~ 6)

다른 숫자를 입력 받으면 다시 입력 받는다!

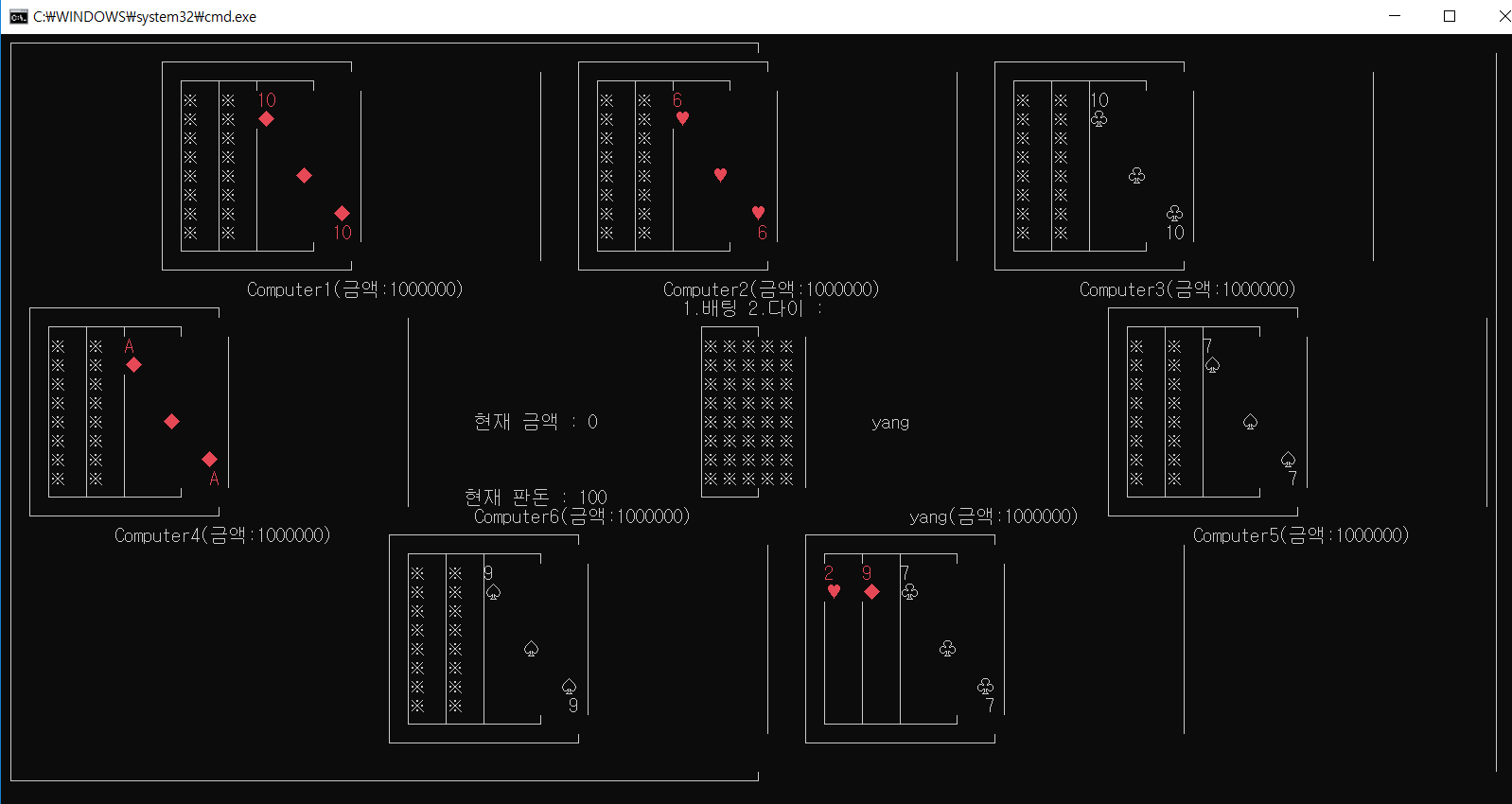
Step 2: 52장의 카드를 랜덤 하게 섞는다. (섞는 과정을 동적으로 보여준다.)

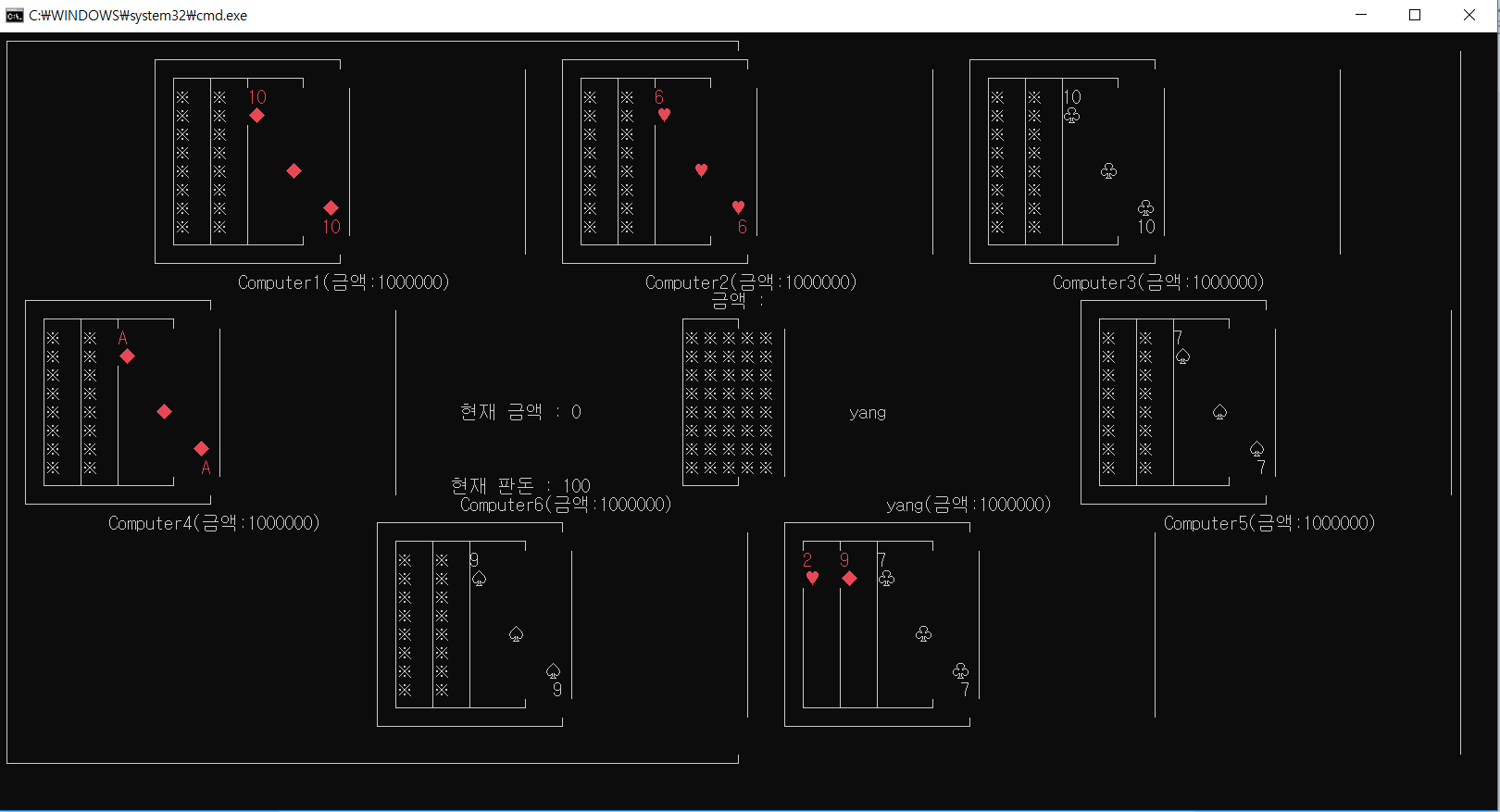


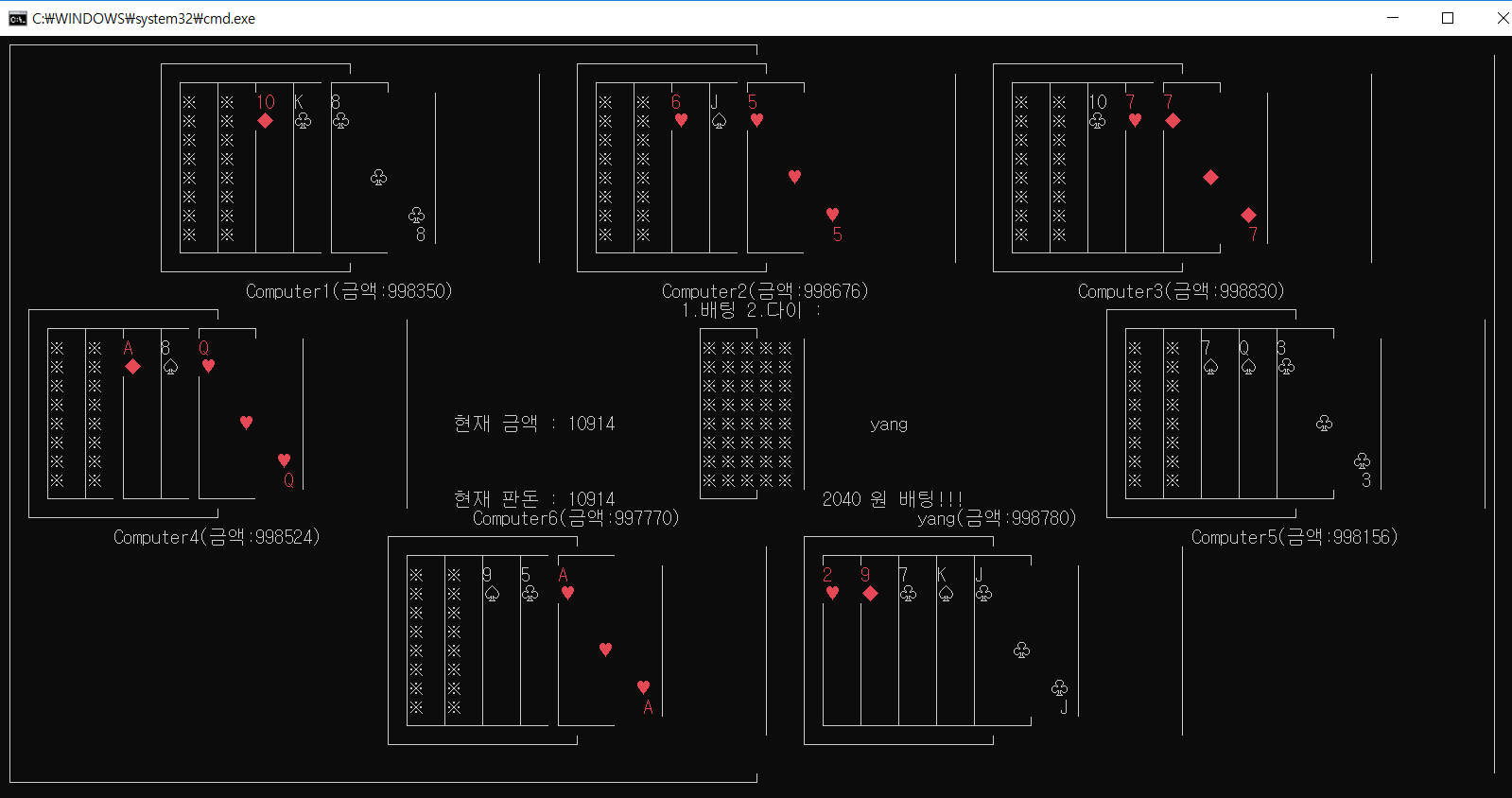
Step 3: 참석자들에게 카드 1장씩 순서대로 3장을 나누어 준다. (이것으로 순서대로 표현한다.)

Step 4: 각자 3장중에 1장씩 오픈한다.

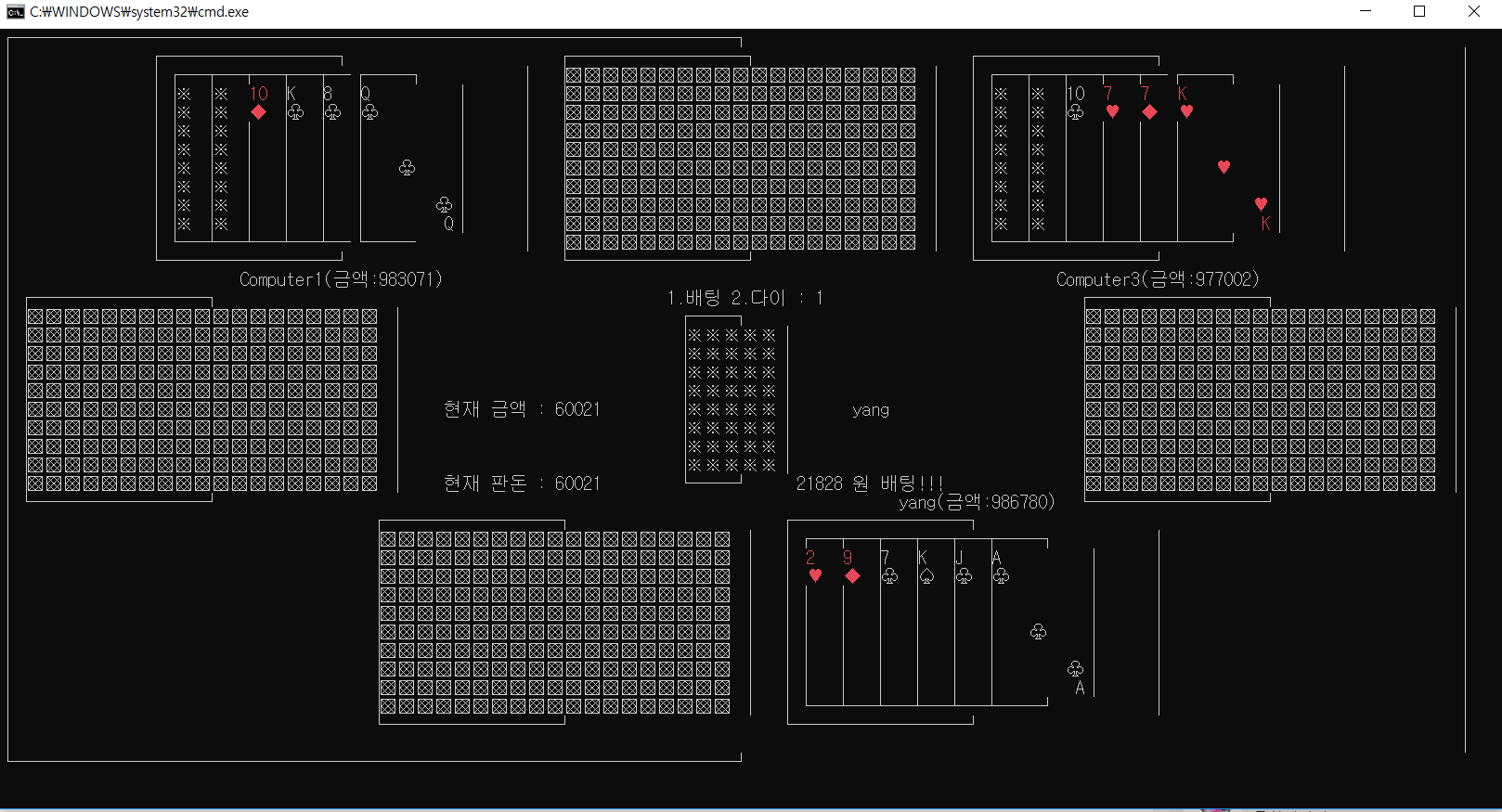
- 오픈하는 카드 선택은 3장의 카드 중 세번째 카드를 오픈한다.

Step 5: 상대방의 카드보다 자신의 카드가 좋지 않다고 판단되면 언제든 경기를 포기하는 다이(Die)를 하거나 베팅할 수 있다. (컴퓨터도 다이할 수 있다.)

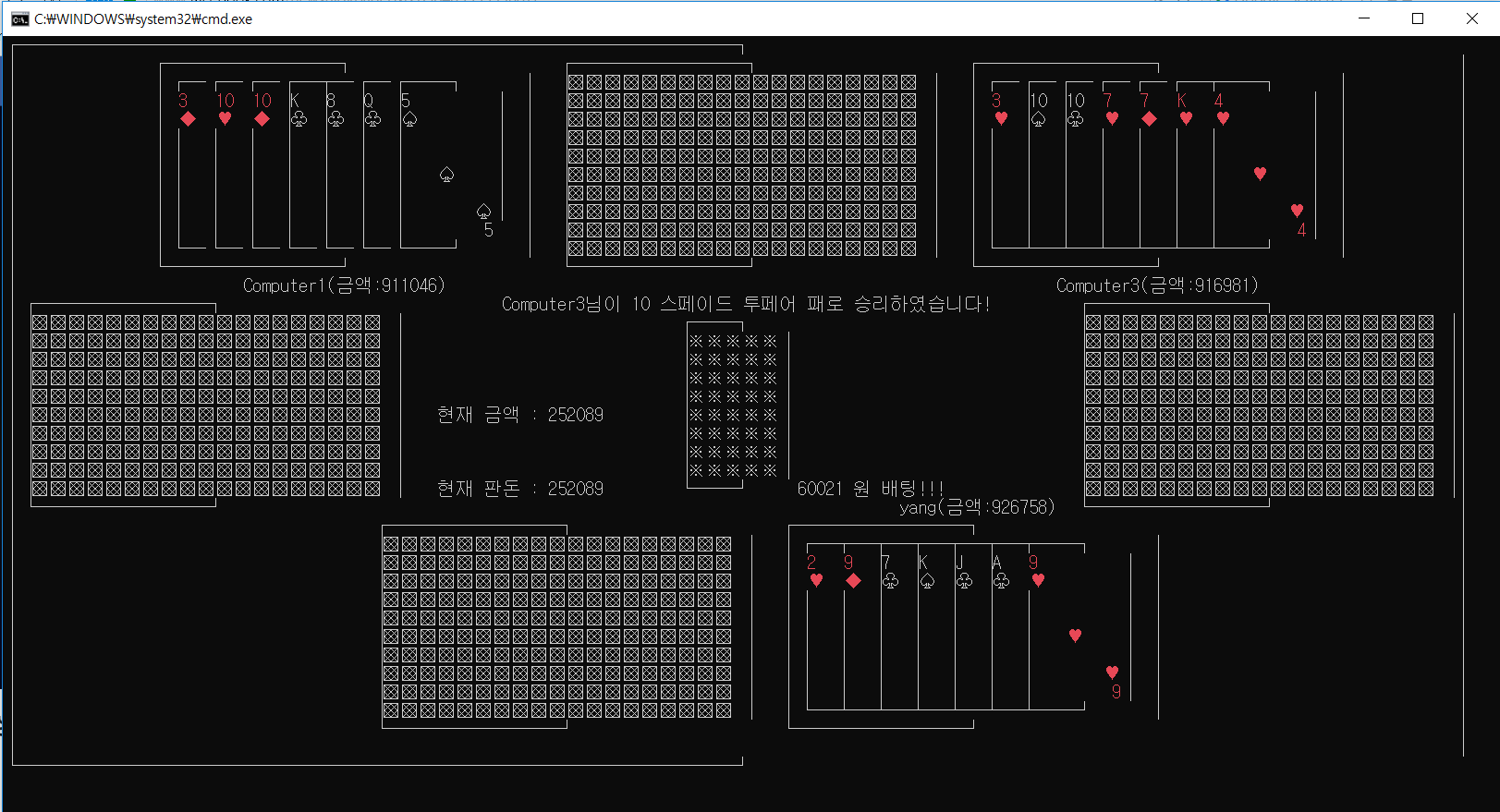
Step 6: 각자 일정 금액을 베팅한다. 금액은 최소 현재 판돈~ 현재 판돈의 2배 사이의 금액을 입력받는다. 그리고 컴퓨터도 돌아가면서 베팅을 한다.

Step 7: 참석자들에게 카드 1장씩 순서대로 나누어 준다.

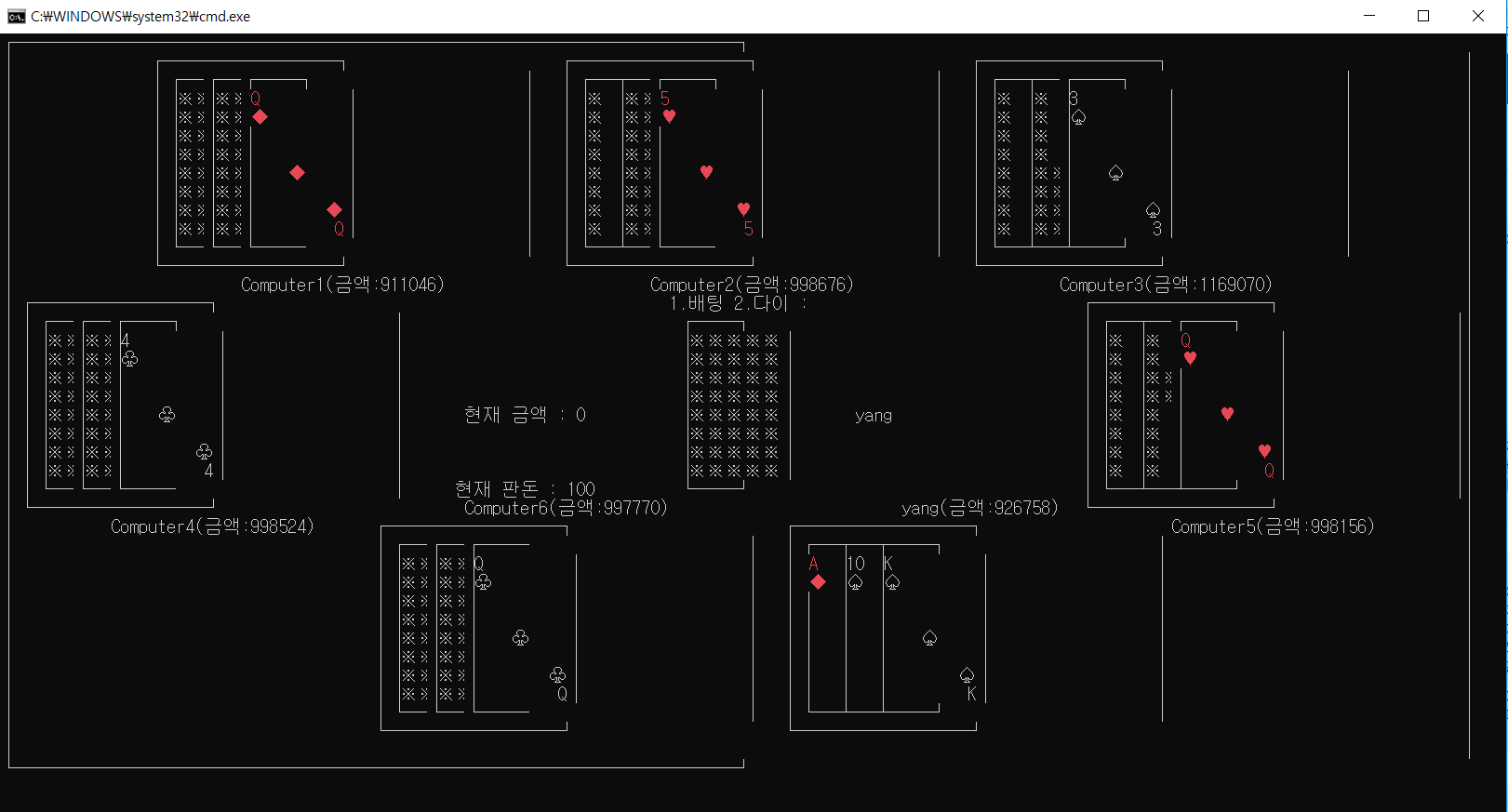
Step 8: 카드를 한 장씩 나누어준 후 일정금액을 베팅을 반복 수행하는 방식으로 최대 7장까지 나누어 주도록 한다.



컴퓨터도 필요하면 다이를 한다.

Step 9: 최종 7장까지 카드가 배분되고, 베팅이 끝났으면 승자를 판단하여 결과를 화면에 보인다.

그리고 이 과정을 반복한다.

Step 10: 이 과정을 반복한다.

이전 게임의 남은 금액을 이어서 계속해서 게임을 한다.