

CSED101. Programming & Problem solving

Fall, 2014

Programming Assignment #2 (60 points)

류승호 (seunghoryu@postech.ac.kr)

■ **Due:** 2014. 10. 16 23:59

■ **Development Environment:** Windows Visual Studio 2010

■ **제출물**

- **C Code file (.c)**
 - 파일이름은 반드시 "**assn2.c**" 로 할 것.
 - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 **주석**을 붙일 것.
- **보고서 파일 (.doc(x) or .hwp)**
 - 보고서는 "**assn2.doc(x)**" 또는 "**assn2.hwp**"로 저장할 것. (보고서는 통합하여 작성)
 - AssnReadMe.pdf 를 참조하여 작성할 것.
 - **프로그램 실행 화면을 캡처하여 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.**

■ Problem: 주사위 게임

(목적)

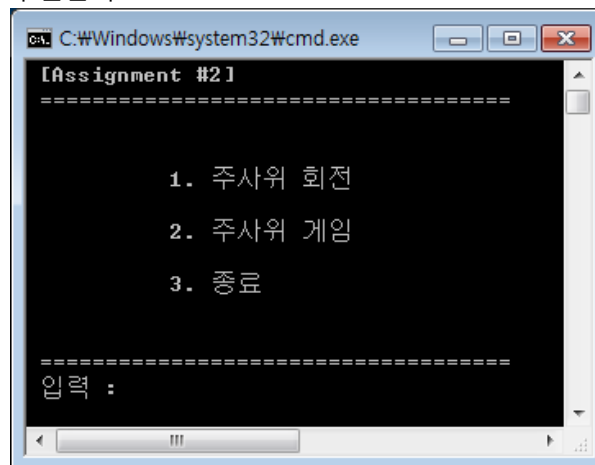
이번 과제를 통하여 조건문, 반복문, 사용자 정의 함수 및 라이브러리 함수 사용법을 익힌다.

(주의사항)

1. 이번 과제는 함수를 정의하고 사용하는 방법을 익히는 문제이므로 사용자 정의 함수를 사용하지 않고 main함수에 모두 기능을 구현한 경우 감점 처리 함. (각 문제의 끝부분에 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인한 후, 구현하도록 한다.)
2. 프로그램 구현 시, main() 함수를 호출하여 사용하지 않는다.
3. 전역 변수는 사용할 수 없으며, 아직 수업시간에 다루지 않은 배열이나 포인터 등은 사용하지 않는다.

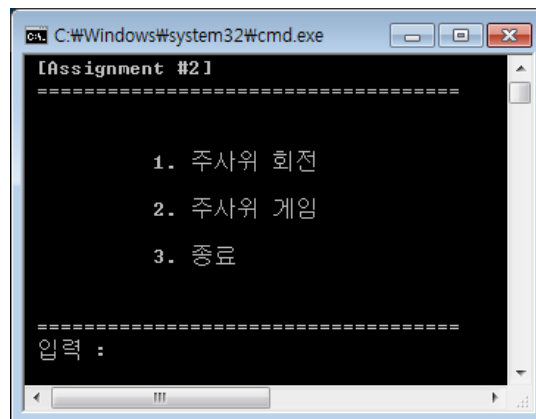
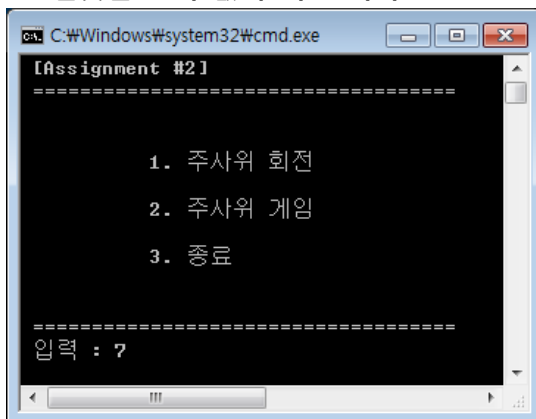
I. (5점) 초기 선택 메뉴

1. 아래처럼 2가지 기능이 가능한 프로그램을 작성하여 사용자로 하여금 원하는 기능을 선택 할 수 있는 '선택 메뉴'를 구현한다.



2. 숫자 1, 2, 3 외의 값이 입력되면, 위의 화면이 그대로 다시 출력되며 재입력을 받는다. 이 부분 구현에 대해서는 아래의 Tips (마지막 쪽)에 설명을 참조한다.

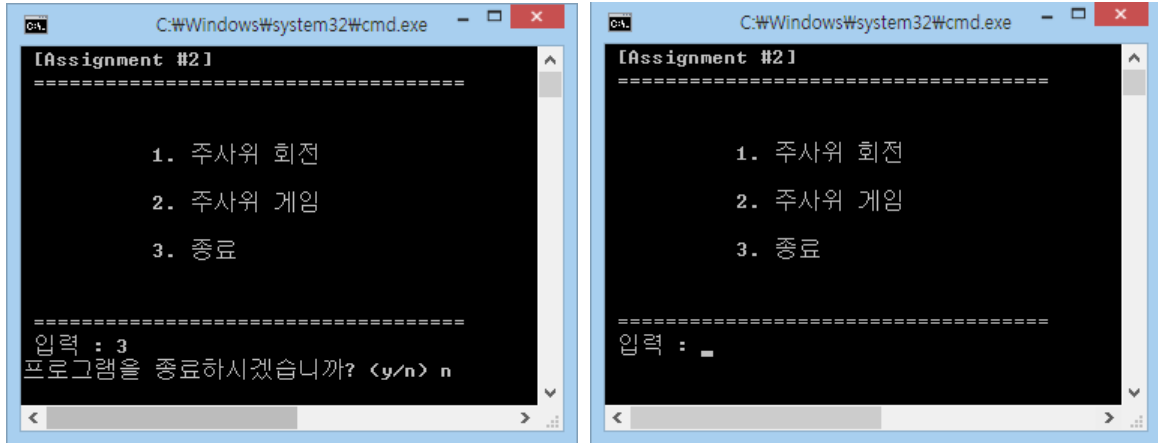
<잘못된 입력 값에 대한 예제>



사용자가 입력범위 외의 값 7을 입력했을 때(왼쪽 실행 화면), 범위 이외의 값이므로 오른쪽 실행 화면이 그대로 다시 출력되어 입력 받을 준비를 한다.

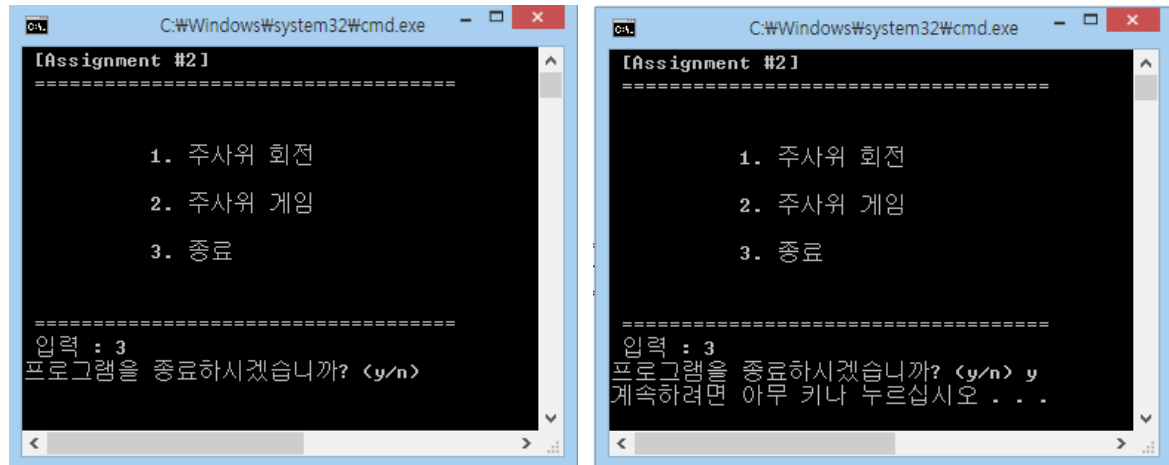
3. 3을 입력하면, 아래의 예제와 같이 프로그램 종료 여부에 대한 질문 "프로그램을 종료하시겠습니까? (y/n)"를 출력하고 입력에 따라서 프로그램을 종료하거나 초기화면으로 돌아간다. y 와 n 이외의 입력 값에 대해서는 고려하지 않는다.

<프로그램을 종료하지 않는 경우에 대한 예제>



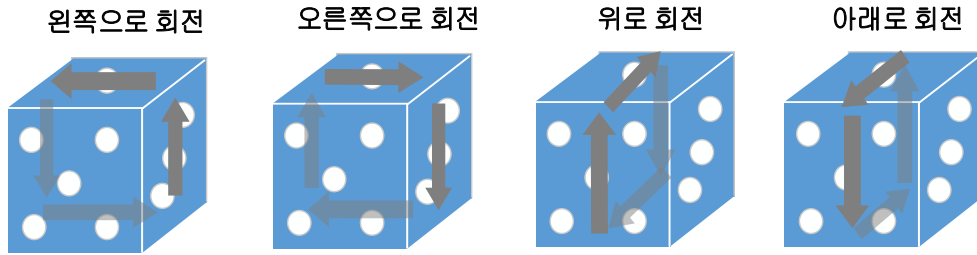
사용자가 'n'을 입력하면 초기 선택 메뉴화면으로 돌아가서 입력을 다시 받는다.

<프로그램을 종료하는 경우에 대한 예제>



사용자가 'y'를 입력하면 프로그램은 종료가 된다.

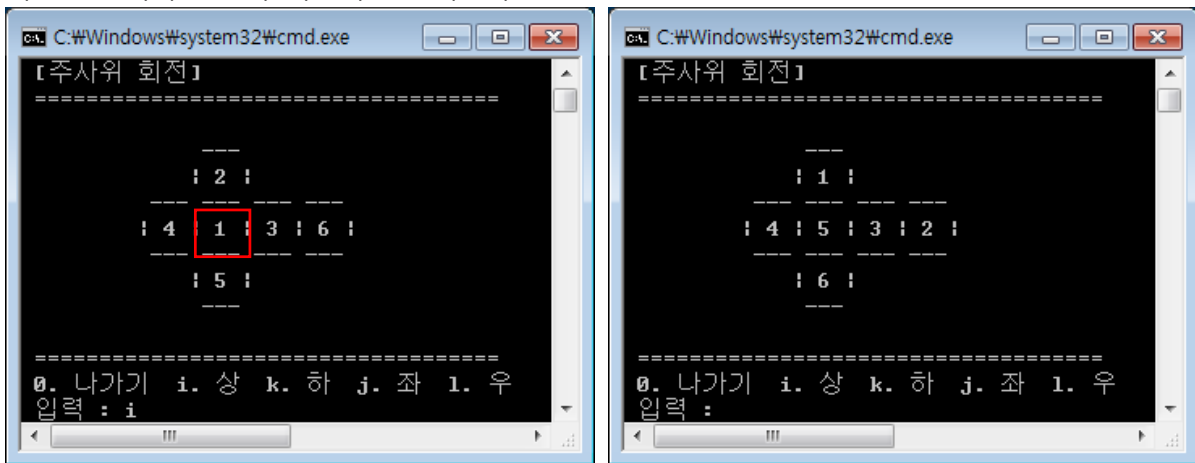
4. 이 문제를 해결하기 위해 반드시 아래의 사용자 정의 함수를 정의하고 사용해야 하며, 아래 함수 외의 필요한 함수를 정의해서 사용할 수 있다.
- **print_menu**: 메뉴화면을 출력하는 함수
 - **select_menu**: 사용자에게 메뉴를 입력 받고 선택 메뉴를 리턴하는 함수



B. 위의 입력 이외에 대해서는 프로그램은 반응하지 않는다. (입력 받기 전과 같은 화면이 다시 출력되며 입력 받을 준비를 한다.)

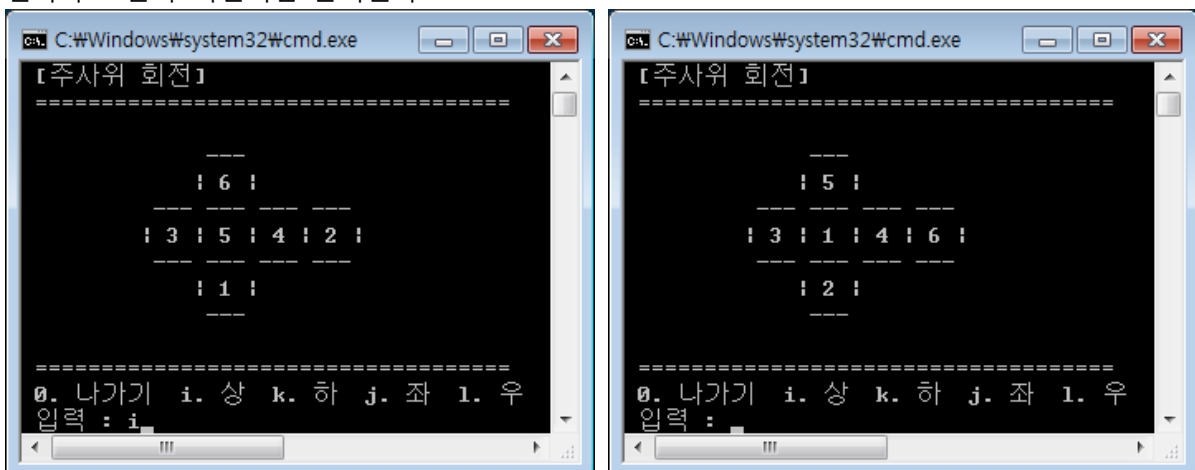
(실행예제 1)

빨간색 사각형은 주사위의 값 (윗면)을 의미하며, 아래의 왼쪽 화면에서 i 를 입력하면 위로 회전한 결과가 오른쪽 화면처럼 출력된다.



(실행예제 2)

아래의 왼쪽 예제는 랜덤하게 주사위 값이 5가 선택된 경우이며, i 를 입력하면 위로 회전한 결과가 오른쪽 화면처럼 출력된다.



3. 이 문제를 해결하기 위해 반드시 아래의 사용자 정의 함수를 정의하고 사용해야 하며, 아래 함수 외의 필요한 함수를 정의해서 사용할 수 있다.

- rotate_dice: 임의의 주사위를 생성하고 사용자의 입력에 따라 주사위 눈의 위치 변경
- print_dice: 주사위 각 면의 값을 위의 예제처럼 화면에 출력

III. (35점) 주사위 게임

(문제)

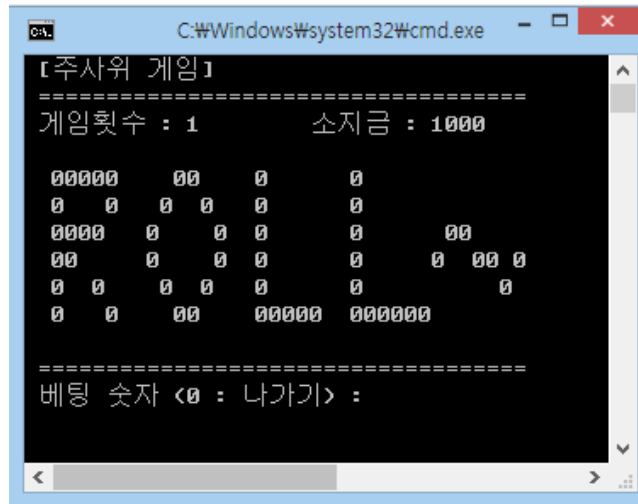
두 개의 주사위를 던졌을 때, 나올 숫자의 합을 예측하여 베팅(betting)을 하는 게임을 작성한다.

(주의사항)

게임 화면의 구성은 각자 자유롭게 하되 게임 플레이어가 게임 횟수, 소지금(게임 머니), 베팅 숫자, 베팅 금액, 베팅 결과를 확인 가능하도록 구성 한다.

(설명)

초기 메뉴에서 2를 입력하여 [주사위 게임]을 선택하면 다음과 같이 '주사위 게임'이 실행된다.

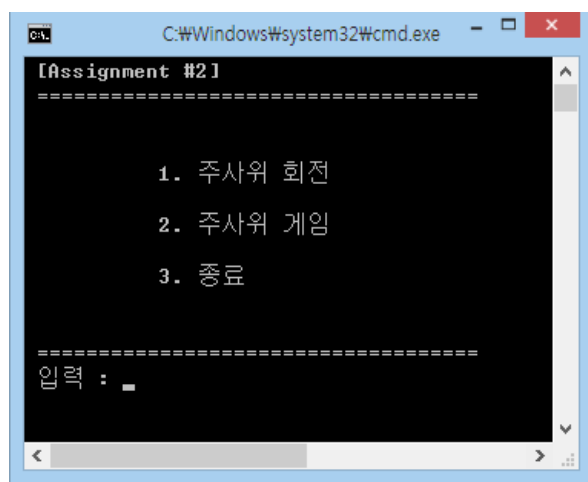
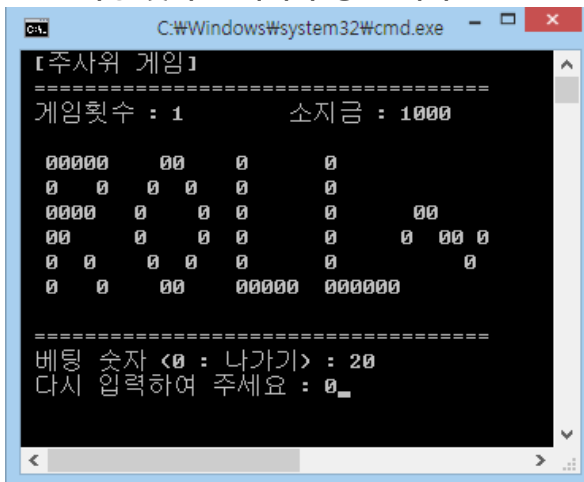


1. **게임**은 베팅과 결과를 확인 하는 과정을 반복하면서 진행된다.

A. 게임 시작 때 소지금(게임 머니) 1000원을 지급받으며, 사용자에게 베팅할 숫자와 베팅 금액을 입력 받는다.

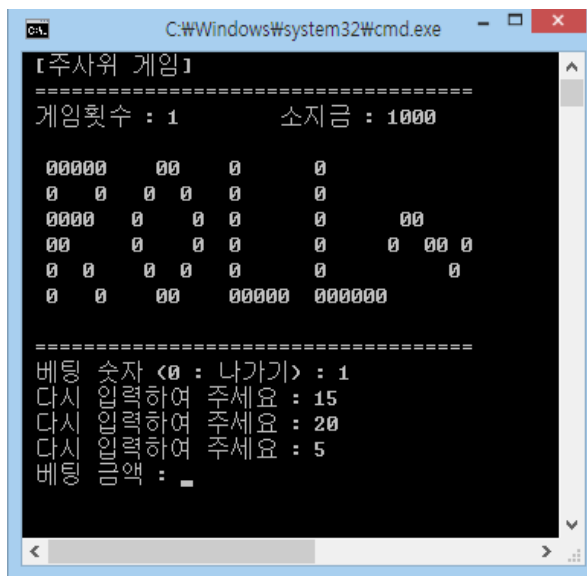
가) 베팅할 숫자는 2 ~ 12 사이이며 이 외의 숫자를 입력하면 아래와 같이 다시 입력을 받는다. (단, 베팅 숫자를 0으로 입력하는 경우는 게임 종료를 의미한다)

<베팅 숫자 입력에서 종료 예제>

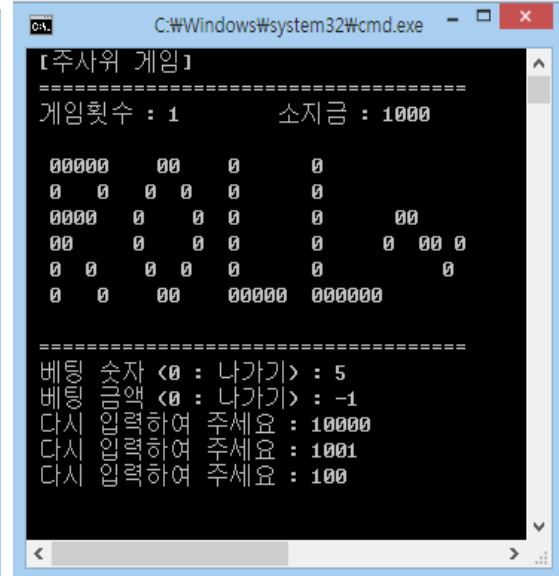


나) 베팅 금액은 소지금액 내에서 1원 단위로 결정 할 수 있다. 베팅 금액이 소지금액을 초과하거나 음수값을 입력하면 아래와 같이 다시 입력을 받는다.
(단, 베팅 금액을 0으로 입력하는 경우는 게임 종료를 의미한다)

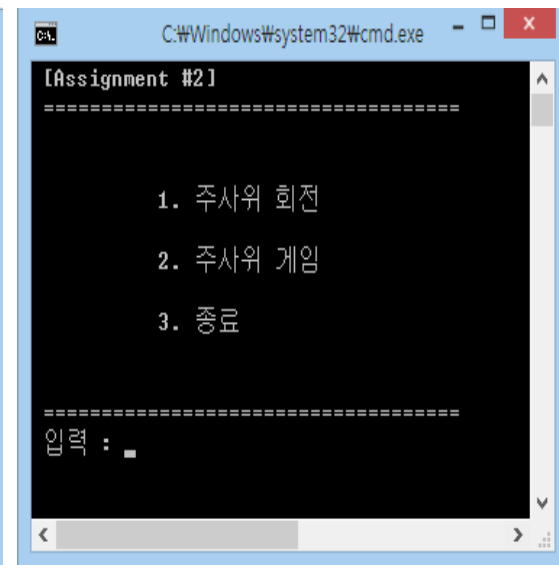
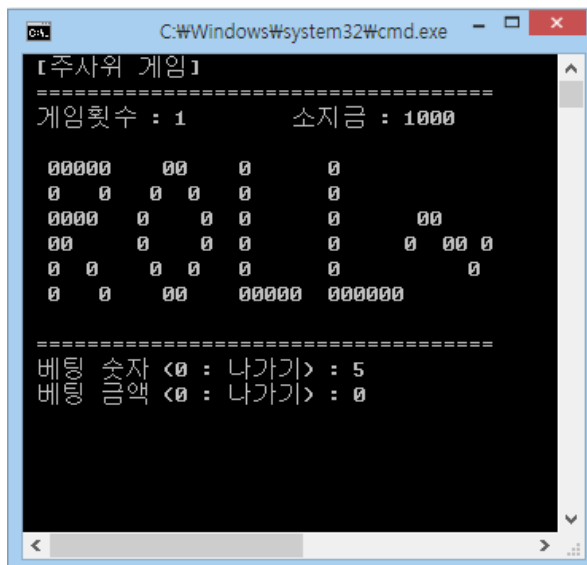
<베팅 숫자 입력 예제>



<베팅 금액 입력 예제>



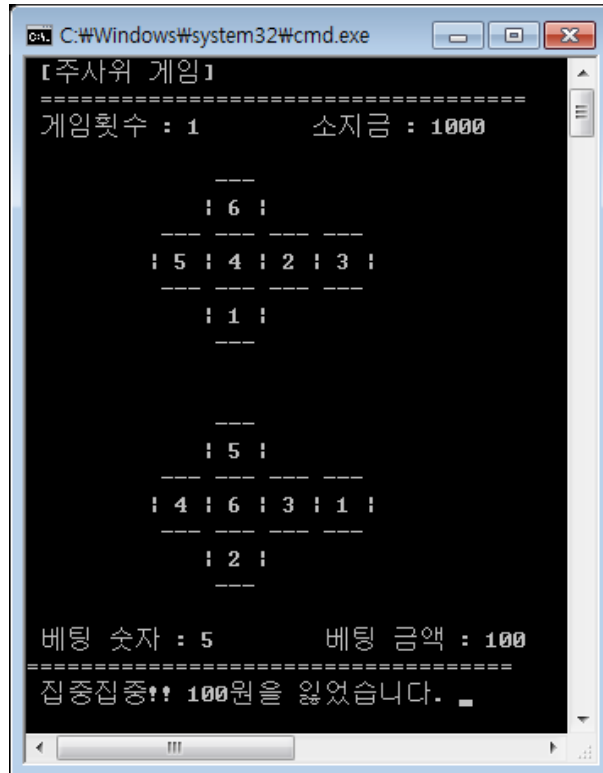
<베팅 금액 입력에서 종료 예제>



B. 베팅 후 아래와 같이 게임횟수, 소지금, 2개의 주사위, 베팅 숫자, 베팅 금액 그리고 베팅 결과를 보여준다.

가) 두 개의 주사위는 컴퓨터가 랜덤하게 값을 선택한다.

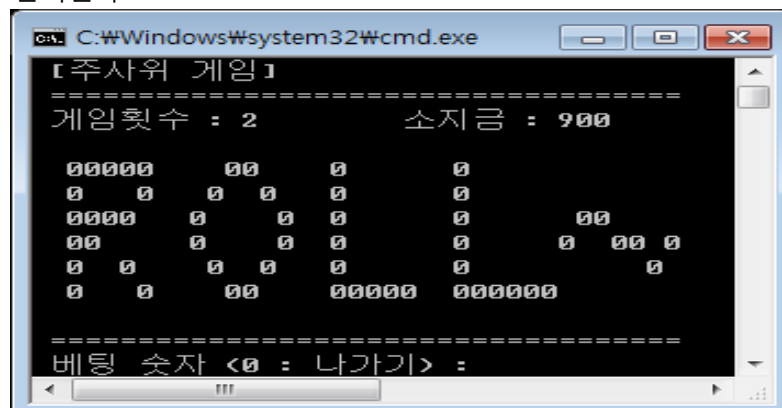
주사위 출력은 [주사위 회전]에서 사용한 주사위 출력 함수 print_dice() 를 사용하여 랜덤한 두 개의 수가 아래와 같이 출력되게 한다.



나) 두 주사위의 숫자 합과 베팅한 숫자의 차이를 기준으로 **보상**을 받는다.

- 컴퓨터가 생성한 2개의 주사위 숫자의 합과 사용자가 베팅한 숫자 차이가 0, 즉 정확히 맞춘 경우 베팅한 금액을 얻음
- 숫자 차이가 1인 경우 베팅한 금액의 절반(50%)을 얻음
- 숫자 차이가 2이 경우 베팅한 금액의 절반을 잃음
(단, 베팅액이 홀수 인 경우 소수점은 올림으로 계산)
- 그 외의 경우는 베팅한 금액을 잃음

C. 위의 베팅 후의 결과 화면에서 엔터를 입력하면 게임횟수와 소지금이 업데이트된 화면이 아래와 같이 출력된다.



<숫자를 맞추는 경우 예제>



<오차 범위 1인 경우 예제>



<오차 범위 2인 경우 예제>

```

C:\Windows\system32\cmd.exe
[주사위 게임]
=====
게임횟수 : 6      소지금 : 955

  ---
  | 6 |
  ---
 | 3 | 5 | 4 | 2 |
  ---
  | 1 |
  ---

  ---
  | 6 |
  ---
 | 4 | 2 | 3 | 5 |
  ---
  | 1 |
  ---

베팅 숫자 : 5      베팅 금액 : 100
=====
좀 더 분발하세요!! 50원을 잃었습니다.
  
```

```

C:\Windows\system32\cmd.exe
[주사위 게임]
=====
게임횟수 : 7      소지금 : 905

00000  00  0  0
0  0  0  0  0  0
0000  0  0  0  0  00
00  0  0  0  0  0  00  0
0  0  0  0  0  0  0
0  0  00  00000  000000

=====
베팅 숫자 <0 : 나가기> : _
  
```

<오차범위 3이상인 경우 예제>

```

C:\Windows\system32\cmd.exe
[주사위 게임]
=====
게임횟수 : 3      소지금 : 1150

  ---
  | 6 |
  ---
 | 3 | 5 | 4 | 2 |
  ---
  | 1 |
  ---

  ---
  | 6 |
  ---
 | 5 | 4 | 2 | 3 |
  ---
  | 1 |
  ---

베팅 숫자 : 5      베팅 금액 : 100
=====
집중집중!! 100원을 잃었습니다.
  
```

```

C:\Windows\system32\cmd.exe
[주사위 게임]
=====
게임횟수 : 4      소지금 : 1050

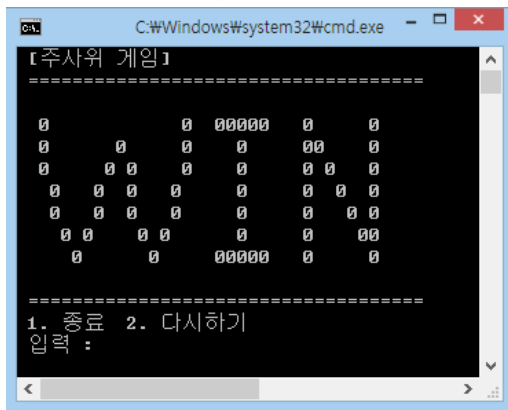
00000  00  0  0
0  0  0  0  0  0
0000  0  0  0  0  00
00  0  0  0  0  0  00  0
0  0  0  0  0  0  0
0  0  00  00000  000000

=====
베팅 숫자 <0 : 나가기> :
  
```

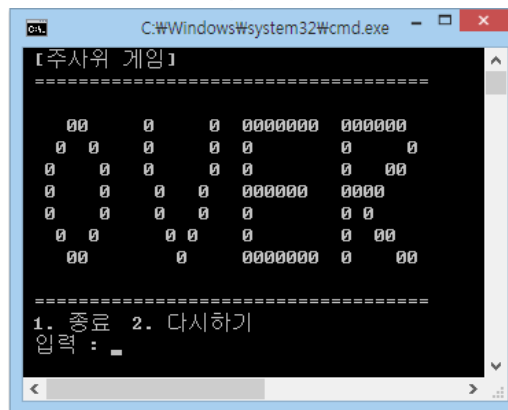
2. **게임이 끝나는 경우**는 다음과 같다.

- A. 처음 지급되는 소지금 1000원을 다 잃은 경우 (게임 패)
- B. 소지금이 3000원을 넘는 경우 (게임 승)
- C. 사용자가 종료를 원하는 경우 (베팅 숫자를 0으로 입력한 경우)
→ 주사위 게임이 종료되고 초기 메뉴 화면이 출력된다.
(main함수 호출을 이용하는 경우 감점 처리)
- D. 승패와 관계없이 게임이 끝난 후, 아래 예시처럼 종료 또는 다시 게임을 할 수 있는 메뉴를 제공한다.
 - 가) '다시하기' 선택 시, 소지금 1000원이 충전되어 게임이 계속 진행되며 게임횟수가 증가한다. 예를 들어 게임횟수 5에서 게임이 끝난 후, '다시하기'를 선택하면 소 지금은 1000원으로 충전되고 게임횟수는 6으로 증가한다.
 - 나) '종료' 선택 시, 처음 초기 메뉴로 돌아간다.

<게임 승 예시 화면>



<게임 패 예시 화면>



3. 이 문제를 해결하기 위해 반드시 아래에서 설명하는 함수를 정의하고 사용해야 하며, 아래 함수 외의 필요한 함수를 정의해서 사용할 수 있다.

- **check_betting_number:** 사용자에게 베팅 숫자를 입력 받고, 입력 받은 숫자가 2부터 12까지에 해당되는 범위일 때 그 숫자를 리턴
- **check_betting_money:** 사용자에게 베팅 금액을 입력 받고, 입력 받은 금액이 소지금을 초과하지 않고 음수가 아닐 때 그 금액을 리턴
- **print_two_dice:** 임의의 값을 가지는 두 개의 주사위 출력 및 주사위의 합을 리턴
- **print_win:** 게임에서 이긴 경우 화면 출력 및 메뉴 제공
- **print_lose:** 게임에서 진 경우 화면 출력 및 메뉴 제공

Tips

- 화면 지우기

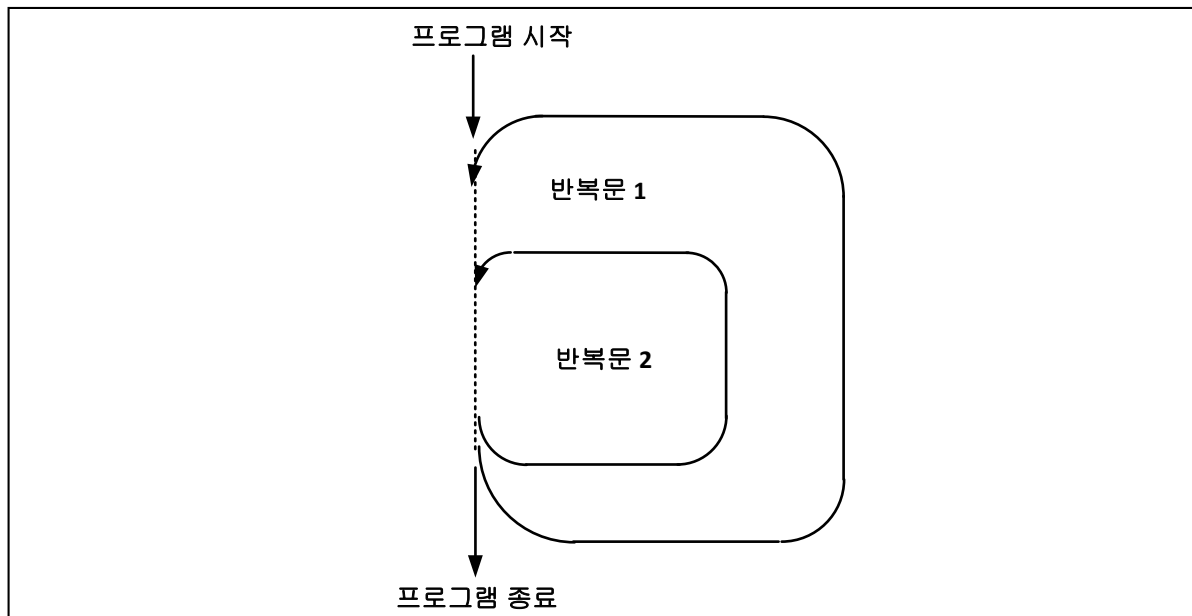
<stdlib.h>를 포함한 뒤 system() 함수를 사용한다.

int system(char*): 명령프롬프트(cmd)상에 정의되어 있는 명령어(dir, cls 등)을 실행하는 함수

```
#include <stdlib.h>

int main()
{
    printf("Hello World\n");
    system("cls"); // 화면에 Hello World 출력 후, 화면을 지움
    ...
}
```

- 사용자 입력 처리



반복문을 중첩하여 화면에 메뉴를 출력한 뒤, 사용자의 입력에 따라서 화면 출력 내용을 바꿀 수 있다. 예를 들어 위의 그림에서 반복문1은 메뉴출력 부분을 담당하고 반복문2는 사용자 입력을 담당한다.

- 주사위 회전, 주사위 게임: 반복문2에서 사용자가 입력을 제대로 할 때 까지 반복
- 초기 메뉴: 반복문2에서 옳지 않은 입력이 들어온 경우, 반복문1로 돌아가서 화면을 새로 출력