과제 1번

< 사용 변수 >

- char 자료형의 i 변수는 5번 시간 출력을 반복하기 위해 사용한 for 문의 제어변수이다. for 문 안에 short 자료형의 year은 연도를, char 자료형의 month, date, hour, min, sec는 순서대로 달, 일, 시, 분, 초를 의미하는 변수이다. monthlist[12] 배열은 char 자료형으로, 달마다 며칠씩 있는지를 저장했으며, int 자료형의 spentsec는 time(NULL)함수를 사용해 1970.01.01 이후 지난 초를 저장하는 변수이다.

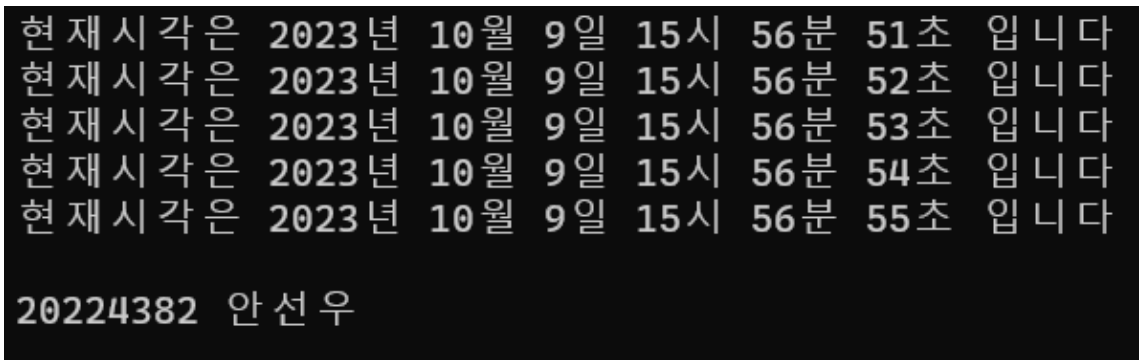
< 풀이 방법 >

[main 함수]

for 문을 사용해 5번 출력되도록 했다. spentsec는 초-분-시-년-월&일 순으로 계산했다. 60을 나눈 나머지를 먼저 sec에 저장하고, sec를 빼고 분 단위로 수정한 후 다시 60으로 나눠 그 나머지를 min에 저장했다. 이후 spentsec를 시간 단위로 수정했는데, 여기서 +9를 통해 UTC+9:00 시간대에 기반하도록 했다. 24를 나눈 나머지를 hour에 저장한 후 spentsec 변수를 날짜 단위로 수정한다. 다음으론 연도-월&일 순으로 지정하는데, 남은 일수가 365보다 작을 때까지 365 또는 366일을 빼주도록 while 문과 if 문을 구성했다. 최종 연도가 윤년이라면 2월을 29일로 계산하도록 monthlist[1]을 29로 수정하는 if 문을 제시하고, 이후로는 다음 monthlist[]보다 작기 전까지 spentsec에서 monthlist[0](1월)부터 차례대로 빼 월, 일을 구한다. 구한 내용을 출력하고 Sleep(1000);을 넣어 1초 이후에 해당 과정을 다시 반복하도록 한다. 2000을 넣는다면 2초 간격으로 시간이 조정되어 출력될 것이다.

< 다른 풀이 방법 >

- <Windows.h> 헤더파일의 Sleep()함수를 사용했으나, clock_t 형 start, end 변수를 사용해 end-start>1000 일 때 continue 하도록 해 시간을 조정할 수도 있다.
- hour 변수에 애초에 9를 저장해둔 뒤, spentsec % 60 한 내용을 더한 후, 만약 hour 값이 24가 넘는다면 24로 나눈 나머지값을 저장하고, spentsec에 다시 24를 돌려줄 수도 있다.



과제 2번

< 사용 변수 > - 모두 char 자료형을 사용했습니다.

- main 함수에서 userboard[25][3], compboard[25][3]은 유저와 컴퓨터의 빙고 판을 저장하는 배열이다. 32 (ASCII 코드 빈칸)으로 모두 초기화했다. userturn, compturn은 유저와 컴퓨터가 선택한 숫자, userbingo, compbingo는 유저와 컴퓨터의 빙고 개수, 그리고 trial은 게임 회차를 의미한다.
- 모든 함수에서 사용된 I, j는 for 문의 제어변수이다. boardmaker 함수에서의 num 변수는 rand()%25+1의 값을 저장한다. 이후 boarchecker 함수에 innum이란 이름의 매개변수로 전달되어 빙고 판에서 index (변수로, innum이 저장될 인덱스의 값을 전달받는다) 전으로 배열에 동일 값이 있는지 확인하는 데 사용된다.
- bingocounter 함수의 count 변수는 빙고 줄에 몇 개의 숫자가 선택되었는지 확인하는 변수이다(만약 빙고라면 count==5이다). bingo 변수는 최종 빙고의 개수를 의미한다.
- smartcomp4 함수의 count 변수는 빙고 줄에 몇 개의 숫자가 선택되었는지 확인하는 변수이며, empty 변수는 빙고 줄에서 선택되지 않은 요소의 인덱스를 저장하는 변수이다.

1	2	3	4	5
...				
21	22	23	24	25

< 풀이 방법 >

[main 함수]

- 변수를 선언한 후, srand(time(NULL))로 rand 값이 계속 다르게 나오도록 했다. boardmaker 함수로 userboard와 compboard를 생성하고, userboard를 출력한 후 do-while 문을 사용해 5 빙고 전까지 빙고 게임을 진행했다. 매 회차의 처음에는 trial 변수를 사용해 회차가 출력되도록 했고, 홀수 회차는 유저가, 짝수 회차는 컴퓨터가 선택하도록 했다.
- (유저 선택) do-while문을 사용해 유저의 선택 값을 userturn 변수에 입력받았다. 이때, inputchecker 함수 반환 값이 0이라면 이전에 선택한 값을 다시 선택한 것이므로 다시 입력받도록 구성했다. eliminate 함수를 통해 userturn 값을 유저와 컴퓨터의 보드에서 제거(괄호를 씌우는 것으로 대체함)하도록 했다.
- (컴퓨터 선택) do-while문을 사용해 inputchecker 함수 반환 값이 0이 아니도록 했다(중복 입력이 아니도록). 여기에선 if-elseif-else 문을 사용해 **컴퓨터가 자신의 상황을 이해하고 유리한 방향으로 숫자를 선택 (↓↓↓)** 하도록 했다. 이후 과정은 유저 선택 경우와 동일하다.

1) 가장 상단의 if에는 (컴퓨터 기준) 가장 첫 번째 회차이며 빙고 판의 정중앙 수가 선택되지 않은 경우, 이를 선택하도록 했다. 이는 정중앙 숫자는 가로세로대각선 모두로도 뻗어나갈 수 있는 수이기 때문이다.

2) 정중앙 수가 이미 선택된 경우거나 4번째 이상의 회차인 경우, smartcomp4 함수를 이용해 한 줄에 3개 이상이 있는 경우, 선택되지 않은 수를 선택하도록 했다. 해당 내용을 이후 나올 조건들보다 위에 위치시킨 이유는 추후 회차에선 사실상 모서리나 정중앙 숫자 기준 십(十)자 위치의 수를 선언하는 것보단 빙고 줄을 완성하는 게 유리하다 생각했기 때문이다. 또한, smartcomp4 함수를 보면 이전 타입(세로→가로→왼위대각선→오른위대각선)의 빙고를 거의 완료한 후 다음 타입의 빙고로 넘어감을 확인할 수 있는데, 이는 너무 많은 것을 고려할 경우, 프로그램이 매우 복잡해지고, 무엇보다 여러 가지 조건을 고려하는 것보단 한 타입의 빙고를 먼저 완성하는 게 유리하다 판단했기 때문이다.

3) 이에 해당하지 않는다면 모서리에 있는 수 4개를 선택하도록 했다. 이는 모서리에 위치한 수는

가로세로대각선으로 겹치는 수이기에 이기기 위해선 정중앙 값 다음으로 중요하다 생각했기 때문이다.

4) 마지막으론 정중앙 수를 기준으로 십(十)자에 위치한 수를 선택하도록 했다. 이는 정중앙을 기준으로 가로세로를 구성하기에 적합하다 생각했기 때문이다.

- 이후엔 결과를 출력하고 컴퓨터/유저의 빙고 판과 빙고 개수를 모두 출력했다.

[boardmaker 함수]

- void 형으로, main 함수에서 userboard/compboard 배열의 포인터 값을 전달받은 후, rand 함수와 boardchecker 함수를 이용해 중복되는 수 없이 빙고 판에 1-25의 수를 저장한다.

[boardchecker 함수]

- char 형으로, boardmaker 함수에서 rand%25+1의 값과 현재 채우고 있는 배열의 인덱스 값을 전달받아 해당 인덱스 전으로 동일한 값이 존재하는지 확인하고 가능 여부를 반환한다.

[printboard 함수]

- void 형으로, userboard/compboard 배열을 전달받아 이를 출력한다. 이 경우, 짝수 회차 종료 시 출력되는데, 이는 매 회차 종료 출력 시 사용자 입장에서 혼란을 느낄 수 있기 때문이다. if 문을 사용한 것은 출력 시의 간격을 맞추기 위함이다.

[inputchecker 함수]

- char 형으로, 유저 혹은 컴퓨터가 선택한 수를 num 변수로 사용하고, 해당 값이 이전에 이미 선택된 값인지 확인하는 함수이다. 이때, 이미 선택된 경우, | 40 | 숫자 | 41 | 의 형태를 가지기에 board[index][1]이 num과 같은 수일 때 [index][0]이 40인지 아닌지 확인한다.

[eliminate 함수]

- void 형으로, 선택된 숫자의 앞 요소[index][0]을 40으로 (ASCII (의미), 뒷 요소 [index][2]를 41로 (ASCII) 의미) 변경한다.

[bingochecker 함수]

- char 형으로 세로-가로-왼위시작대각선-오른위시작대각선 의 빙고 여부를 확인한다. count 변수를 통해 해당 빙고 줄에 몇 개가 선택되었는지 확인, 이후 count==5라면 bingo로 간주한다.

[smartcomp4 함수]

- char 형으로, bingochecker 함수처럼 빙고 줄에 빈칸의 개수를 확인하고, 그중 하나의 인덱스를 반환하는 함수이다. 한 줄에 3개 이상이 있다면 나머지 값 중 하나를 부른다를 구현한 함수로, empty 변수에 해당 빙고 줄의 빈칸 중 가장 아래/오른쪽의 인덱스를 저장한다. count가 3~4개일 때 empty를 반환하고, 만약 해당 타입(세로/가로/대각선)의 빙고가 모두 존재한다면 다음 타입의 빙고를 완성하기 위해 넘어간다.

< 다른 풀이 방법 >

- compbingo와 userbingo 변수를 사용하지 않아도 된다. 그러나, 동일한 값을 사용하기 위해 계속 bingochecker 함수를 시행하는 게 더 비합리적이라 생각했기에 변수를 만들었다. 전역변수로 만들 수도 있었으나, 변수 줄의 위치 등도 함께 저장해야 하기에 비합리적이라 생각했다.
- 애초에 빙고 판의 모든 내용을 문자열로 선언할 수도 있다. '26', "(26)" 이런 식으로 저장한다면 출력 시에는 편리하나, 사실상 중복된 내용인지 확인하는 등의 작업이 매우 복잡해진다. 그래서 ASCII 코드를 사용해 모두 숫자로 처리했다.

```
유저의 빙고판입니다.
     22      14       6      23       4
     12      11      10       5       8
     24       7       9       2      15
     17      25      21      20      13
      1      16      18       3      19

1번째 판
유저 순서입니다. 원하는 번호를 입력하세요. 9

2번째 판
컴퓨터는 25를 골랐습니다.
     22      14       6      23       4
     12      11      10       5       8
     24       7    ( 9)       2      15
     17    (25)      21      20      13
      1      16      18       3      19
  빙고의 수: 0

… (중략) …

18번째 판
컴퓨터는 7를 골랐습니다.
   (22)      14    ( 6)      23       4
    (12)    (11)    (10)    ( 5)    ( 8)
    (24)    ( 7)    ( 9)       2      15
     17     (25)    (21)    (20)      13
    ( 1)    (16)    (18)    ( 3)    (19)
  빙고의 수: 4

19번째 판
유저 순서입니다. 원하는 번호를 입력하세요. 17

 유저 승리!
컴퓨터의 빙고판은 이렇습니다.
    ( 9)     (24)      14     (11)      23
    ( 8)     (21)    (12)       2     ( 6)
    (18)     (19)    (25)     ( 7)    (17)
    ( 1)     (10)      15     ( 5)       4
    (16)     (22)      13     ( 3)    (20)
  빙고의 수: 4
유저의 빙고판은 이렇습니다.
   (22)      14     ( 6)      23       4
    (12)    (11)    (10)    ( 5)    ( 8)
    (24)    ( 7)    ( 9)       2      15
    (17)    (25)    (21)    (20)      13
    ( 1)    (16)    (18)    ( 3)    (19)
  빙고의 수: 5

20224382 안선우
```

유저 승리

```
유저의 빙고판입니다.
      6      17       3      22       4
     15      13      14       5      23
      9      19       7      10       8
      1       2      20      21      16
     11      12      18      24      25

1번째 판
유저 순서입니다. 원하는 번호를 입력하세요. 7

2번째 판
컴퓨터는 20를 골랐습니다.
      6      17       3      22       4
     15      13      14       5      23
      9      19    ( 7)      10       8
      1       2    (20)      21      16
     11      12      18      24      25
  빙고의 수: 0

… (중략) …

20번째 판
컴퓨터는 23를 골랐습니다.
    ( 6)    (17)    ( 3)    (22)    ( 4)
      15    (13)    (14)    ( 5)    (23)
     ( 9)    (19)    ( 7)      10    ( 8)
     ( 1)       2    (20)      21    (16)
    (11)      12      18     (24)    (25)
  빙고의 수: 3

21번째 판
유저 순서입니다. 원하는 번호를 입력하세요. 15

 비겼습니다.
컴퓨터의 빙고판은 이렇습니다.
    (11)       2    ( 6)    ( 1)    (22)
     (17)    (16)    ( 8)    ( 4)     10
     (13)    (15)    (20)    (25)    ( 3)
     (18)      21      12    ( 7)    (24)
     (14)    ( 5)    (23)    (19)    ( 9)
  빙고의 수: 5
유저의 빙고판은 이렇습니다.
    ( 6)    (17)    ( 3)    (22)    ( 4)
     (15)    (13)    (14)    ( 5)    (23)
     ( 9)    (19)    ( 7)      10    ( 8)
     ( 1)       2    (20)      21    (16)
    (11)      12      18     (24)    (25)
  빙고의 수: 5

20224382 안선우
```

비김

```
유저의 빙고판입니다.
     21      22       8       4       1
      2      25       7      12      23
      5      16      18      17      14
      9      13      19      11       3
     24       6      10      15      20

1번째 판
유저 순서입니다. 원하는 번호를 입력하세요. 21

2번째 판
컴퓨터는 4를 골랐습니다.
   (21)      22       8    ( 4)       1
      2      25       7      12      23
      5      16      18      17      14
      9      13      19      11       3
     24       6      10      15      20
  빙고의 수: 0

… (중략) …

18번째 판
컴퓨터는 2를 골랐습니다.
   (21)    (22)    ( 8)    ( 4)    ( 1)
    ( 2)    (25)       7    (12)    (23)
    ( 5)     16     (18)    (17)    (14)
      9    (13)      19      11       3
   (24)       6    (10)    (15)    (20)
  빙고의 수: 2

19번째 판
유저 순서입니다. 원하는 번호를 입력하세요. 10
이전에 나온 값입니다. 다시 입력하세요.
유저 순서입니다. 원하는 번호를 입력하세요. 3

 컴퓨터 승리!
컴퓨터의 빙고판은 이렇습니다.
    (21)    (13)    ( 3)       9    ( 5)
      16    ( 1)    (15)      19    (23)
      11    (17)    ( 4)    (24)    (18)
     (10)    (20)    ( 8)    (22)    (25)
       6    (14)    ( 2)       7    (12)
  빙고의 수: 5
유저의 빙고판은 이렇습니다.
    (21)    (22)    ( 8)    ( 4)    ( 1)
     ( 2)    (25)       7    (12)    (23)
     ( 5)     16     (18)    (17)    (14)
       9    (13)      19      11    ( 3)
    (24)       6    (10)    (15)    (20)
  빙고의 수: 3

20224382 안선우
```

컴퓨터 승리 & 중복 선택 시