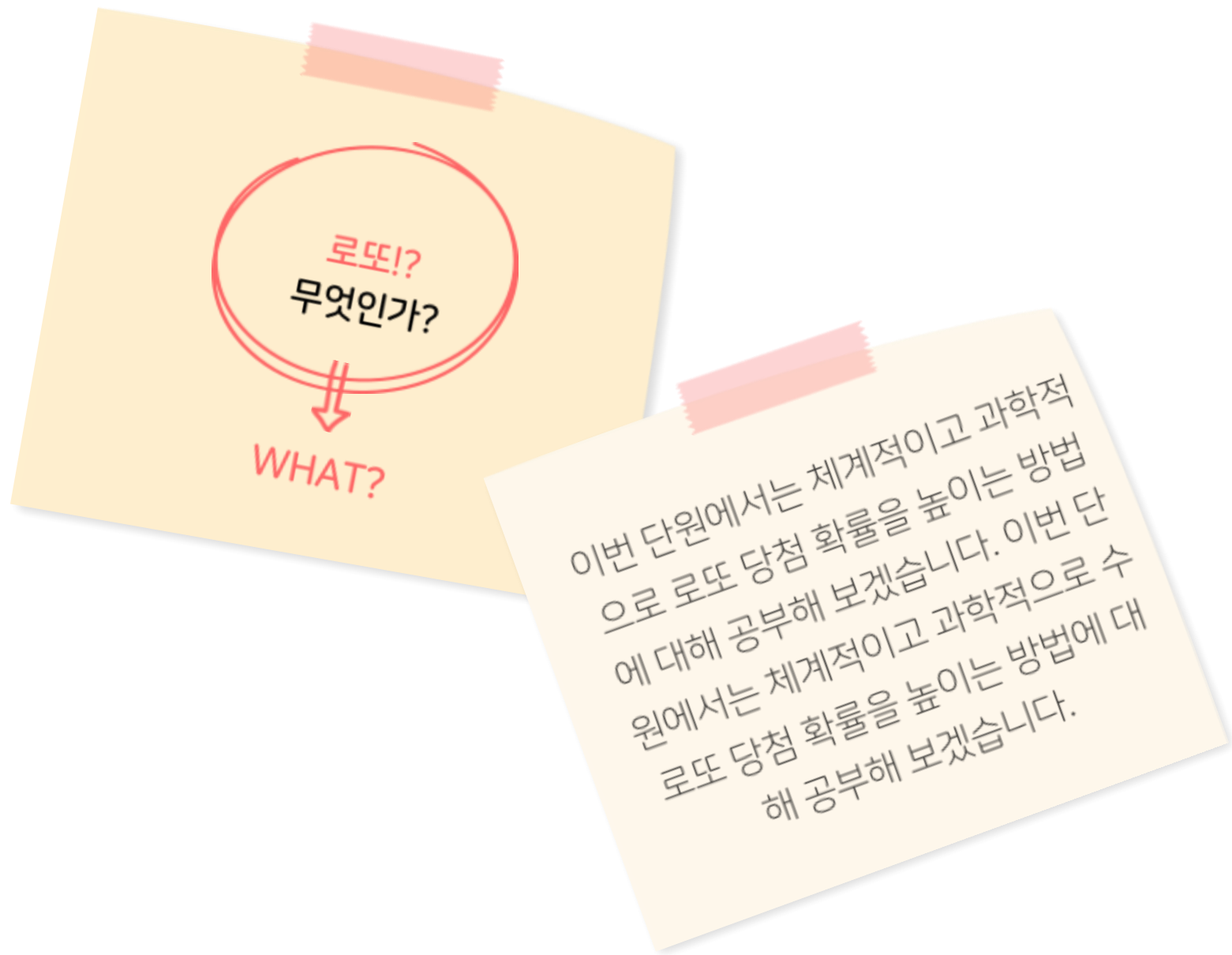


심화 로또의 정석[®]

로또 1등 당첨 확률 높이기!

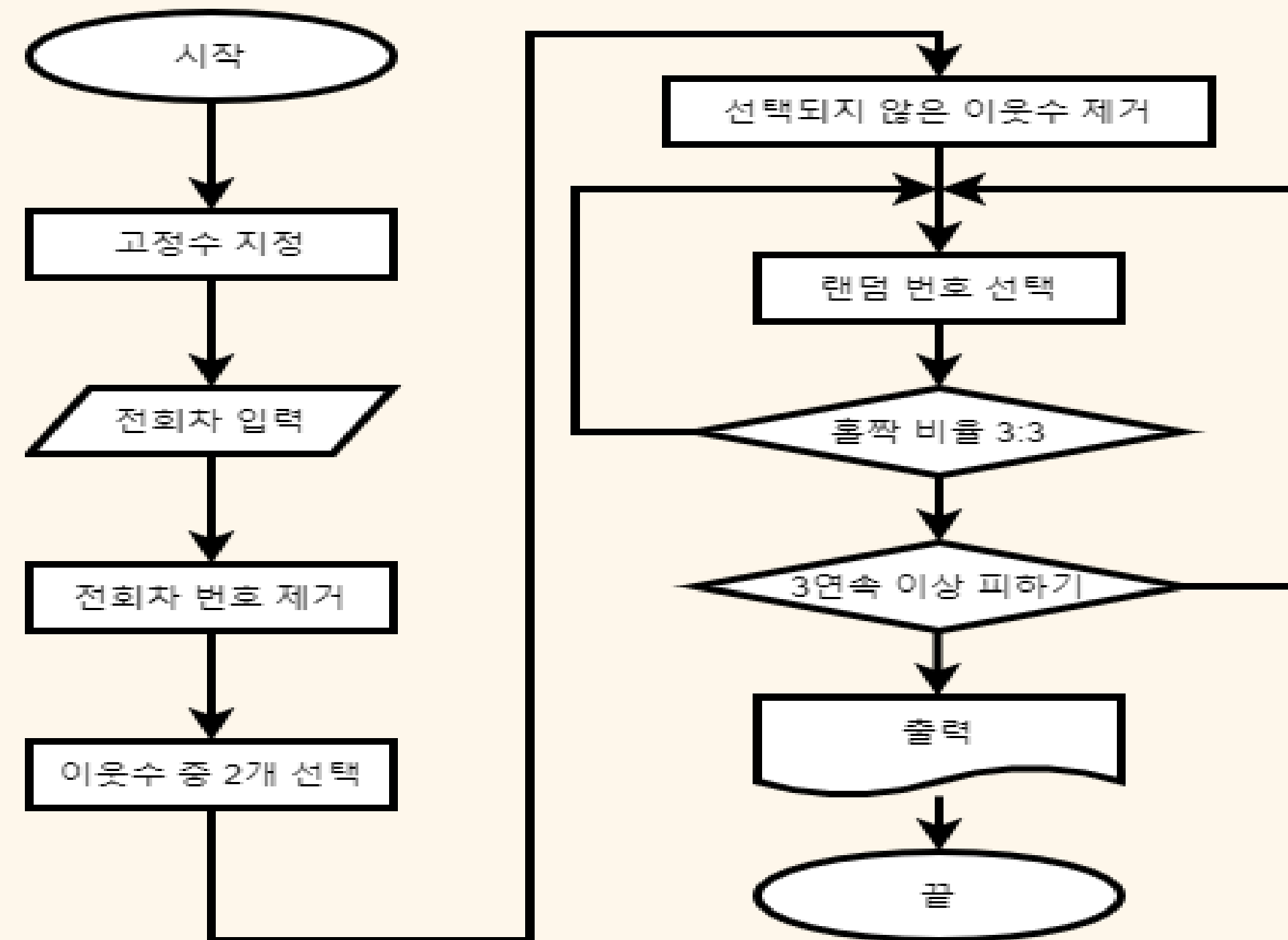
E 조 발표

차례[®]



시도한 목차	no.
고정수 지정	02
전회차 피하기	03
이웃 수	04
홀짝 비율	05
연속 번호 피하기	06

순서도



•

최다 당첨
번호!

-12-

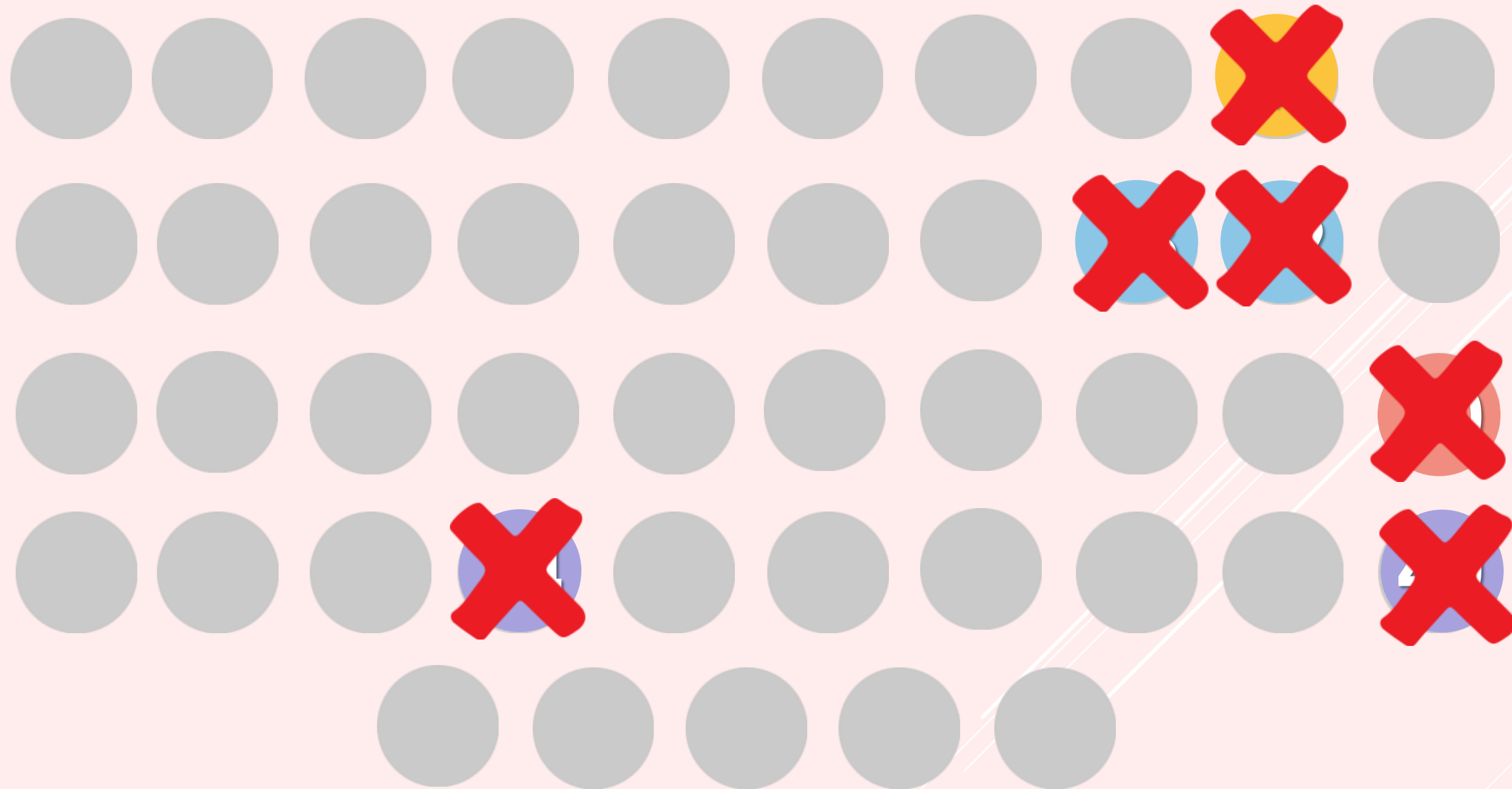
•

현재 사용중인
기계!

733 ~ 946 회차까지의 통계

배열 번호 : 0 1 2 3 4 5 6 7 8 9 10 11
저장된 값 : 1 2 3 4 5 6 7 8 9 10 11 12

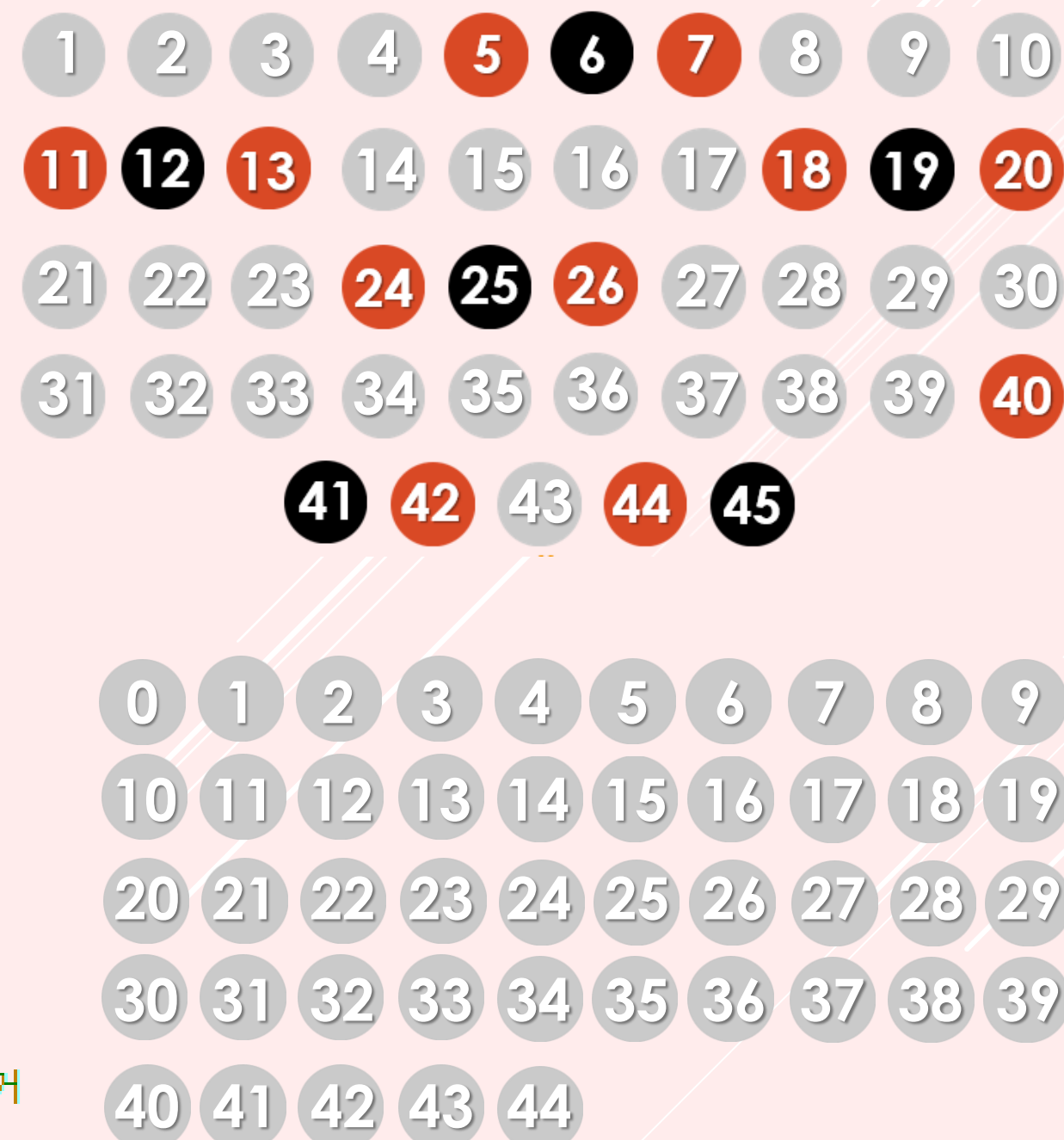
```
47 int arr[45];  
48 int fixnum;  
49 int lastnum[6], adnum[12]; //전화차 당첨번호, 인접값  
50 int lotto[6]; //이번주 예상번호  
51 int i, evencnt = 0, oddcnt = 0;  
52 int rannum, token = 0;  
53 srand(time(NULL)); // rand함수 시간단위 난수로 초기화  
54  
55 for (i = 0; i < 45; i++)  
56     arr[i] = i + 1;  
57  
58 fixnum = lotto[0] = 12; //고정번호 : 최다빈출  
59 arr[fixnum - 1] = 0;
```



```

67 for (i = 0; i < 6; i++)
68 {
69     scanf("%d", &lastnum[i]); //전화차 번호입력(오름차순으로 입력)
70
71
72     if (lastnum[i] == 1) {
73         adnum[i * 2] = 0;
74         adnum[i * 2 + 1] = 2;
75     }
76
77     else if (lastnum[i] == 45) {
78         adnum[i * 2] = 44;
79         adnum[i * 2 + 1] = 0;
80     }
81
82     else {
83         if (lastnum[i] - 1 == lotto[0]) //고정값과 같을 경우 제거
84             adnum[i * 2] = 0;
85         else
86             adnum[i * 2] = lastnum[i] - 1; //인접값 구하기
87
88         if (lastnum[i] + 1 == lotto[0])
89             adnum[i * 2 + 1] = 0;
90         else
91             adnum[i * 2 + 1] = lastnum[i] + 1;
92
93         if (i != 0 && adnum[i * 2 - 1] > adnum[i * 2]) { //전화차 번호와 같은 경우 제거
94             adnum[i * 2 - 1] = adnum[i * 2] = 0;
95         }
96     }
97     arr[lastnum[i] - 1] = 0; //전화차 번호제외
98 }

```



이웃수를 선택하라!

84.2%의 확률!

바로 전 회차에 출연했었던 당첨 번호들이 그 다음 추첨에서 대량으로 출현한 적이 있었다. 그리고 이때 이를 '이웃수'의 강세라고 언급한 적이 있다.

지금까지 이웃수가 세 개 이상 출현한 사례는 총 83회로 전체의 13% 수준, 세 개 출현이 67회, 네 개 출현이 회로 상당히 꾸준한 출현을 기록하는 모습을 보였다.

19

20

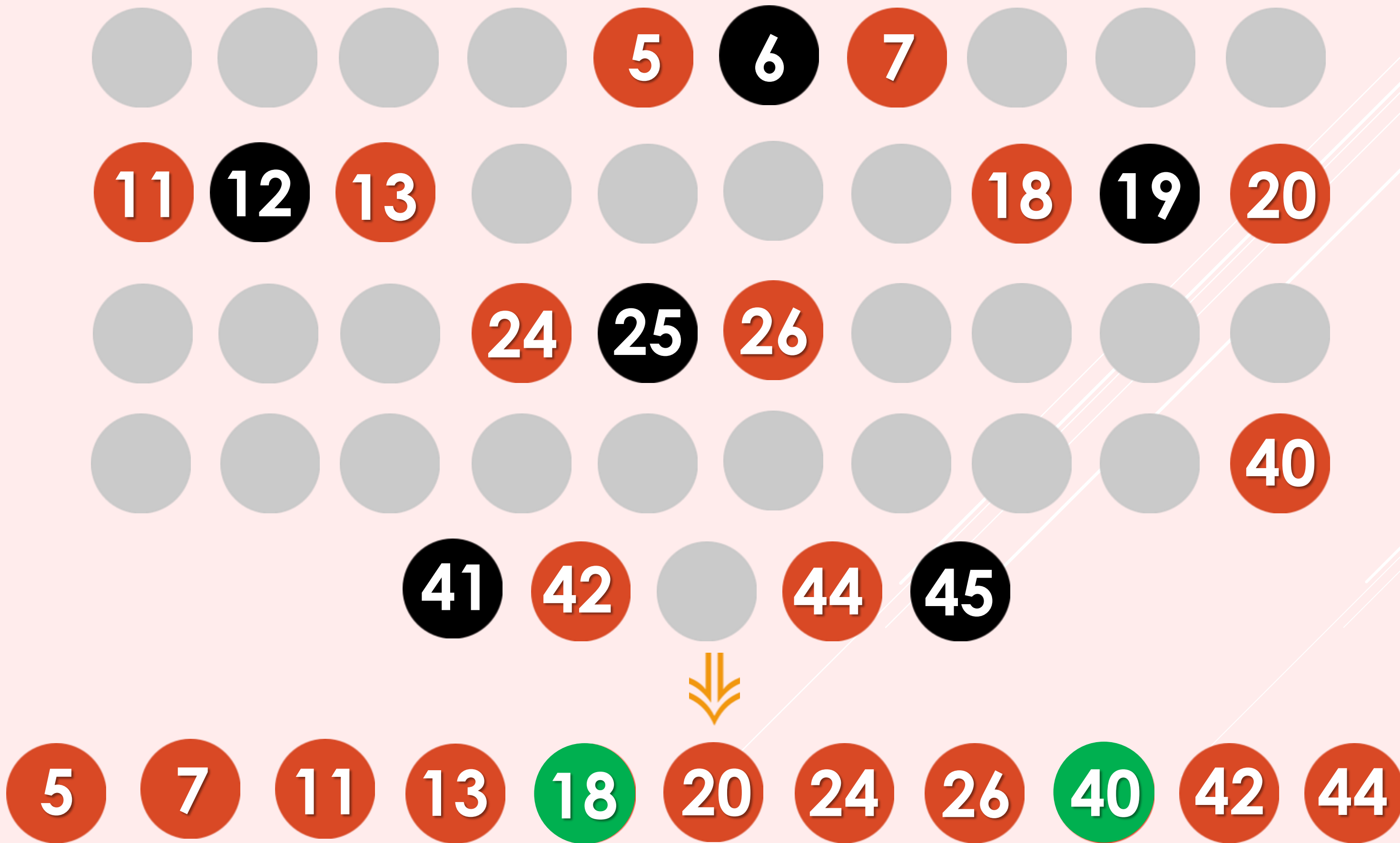
21

84.2%

이웃수 출현

15.8%

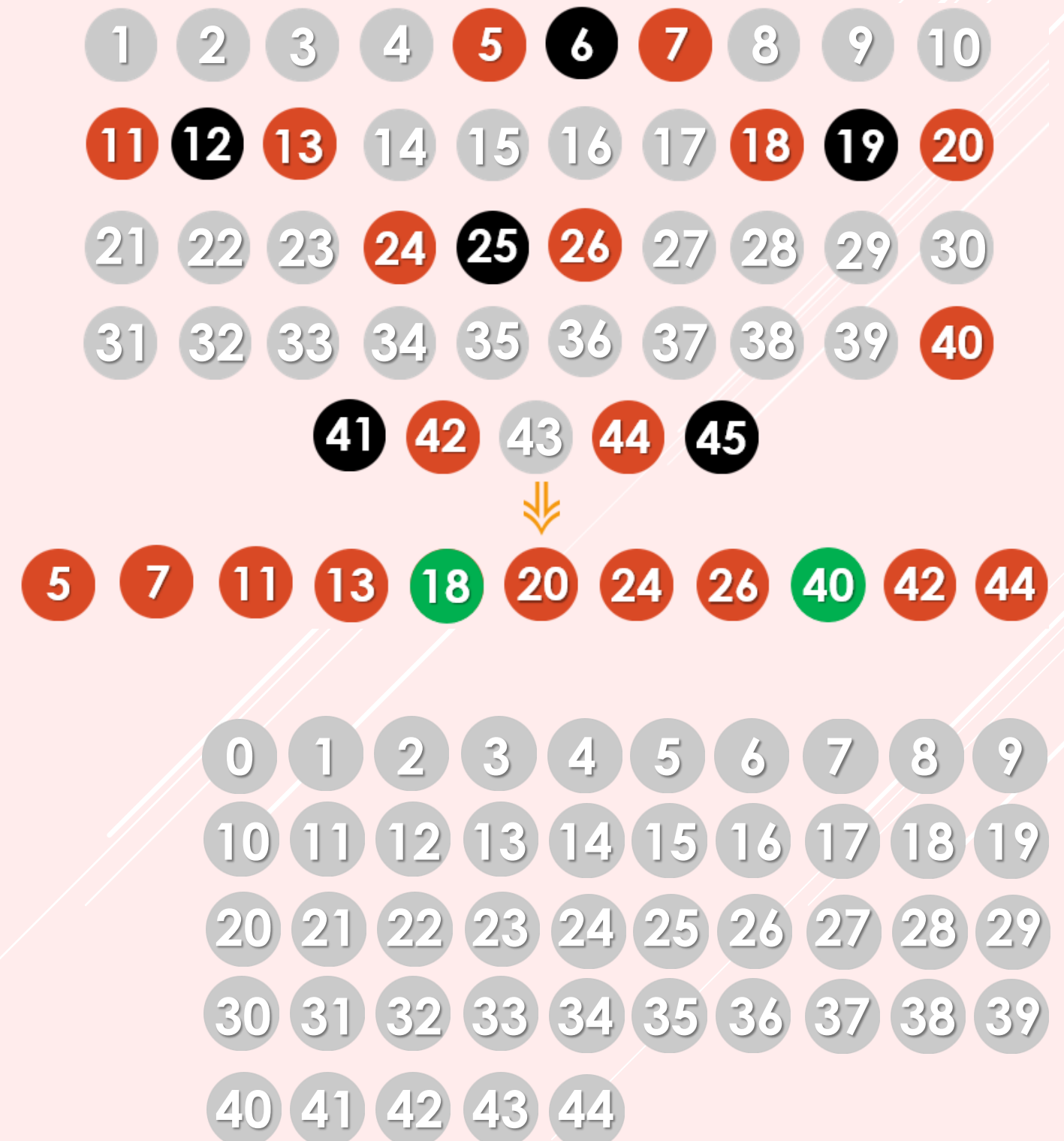
이웃수 비출현



```

67 for (i = 0; i < 6; i++)
68 {
69     scanf("%d", &lastnum[i]); //전화차 번호입력(오름차순으로 입력)
70
71
72     if (lastnum[i] == 1) {
73         adnum[i + 2] = 0;
74         adnum[i + 2 + 1] = 2;
75     }
76
77     else if (lastnum[i] == 45) {
78         adnum[i + 2] = 44;
79         adnum[i + 2 + 1] = 0;
80     }
81
82     else {
83         if (lastnum[i] - 1 == lotto[0]) //고정값과 같을 경우 제거
84             adnum[i + 2] = 0;
85         else
86             adnum[i + 2] = lastnum[i] - 1; //인접값 구하기
87
88         if (lastnum[i] + 1 == lotto[0])
89             adnum[i + 2 + 1] = 0;
90         else
91             adnum[i + 2 + 1] = lastnum[i] + 1;
92
93         if (i != 0 && adnum[i + 2 - 1] > adnum[i + 2]) { //전화차 번호와 같은 경우 제거
94             adnum[i + 2 - 1] = adnum[i + 2] = 0;
95         }
96     }
97     arr[lastnum[i] - 1] = 0; //전화차 번호제외
98 }
99
100 do
101 {
102     lotto[1] = adnum[rand() % 12]; //인접값 중 랜덤 선택
103     lotto[2] = adnum[rand() % 12];
104 } while (lotto[1] == 0 || lotto[2] == 0 || lotto[1] == lotto[2]); //중복제거
105
106 for (i = 0; i < 12; i++) { //인접값제외
107     if (adnum[i] != 0)
108         arr[adnum[i] - 1] = 0;
109 }
110

```



홀짝의 비율을 정해라!

3 : 3

빈도 계산 결과 홀짝 비율 3 : 3 압도적으로 큰 통계를 확인했다.

회차	홀짝 비율	홀6 : 짝0	5 : 1	4 : 2	3 : 3	2 : 4	1 : 5	홀0 : 짝6
소계	-	3회	22회	48회	75회	44회	17회	5회
946	홀2 : 짝4					★		
945	홀4 : 짝2			★				
944	홀3 : 짝3				★			
943	홀3 : 짝3				★			
942	홀2 : 짝4					★		
941	홀3 : 짝3				★			
940	홀3 : 짝3				★			

로또팟 < 리얼 로또 통계 자료 >

6:0 2%

5:1 12%

4:2 18%

3:3 32%

2:4 24%

1:5 8%

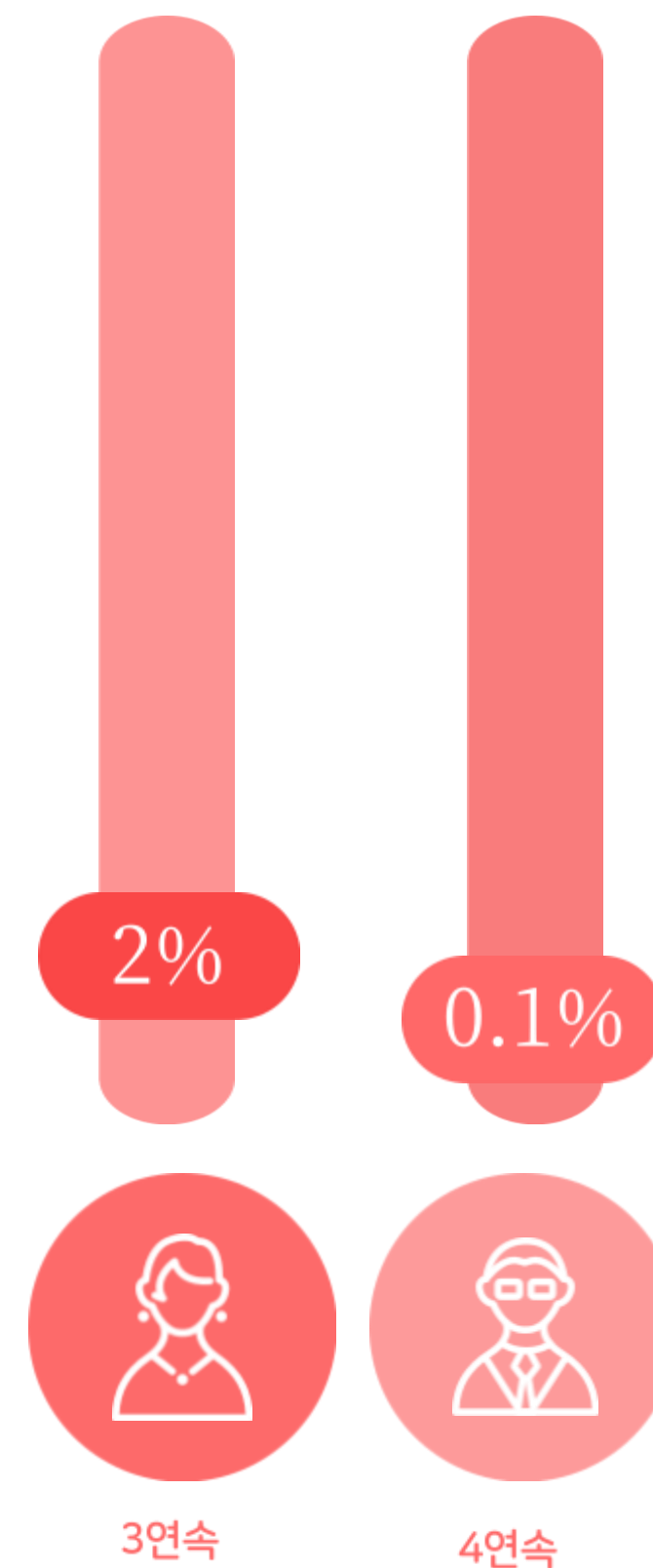
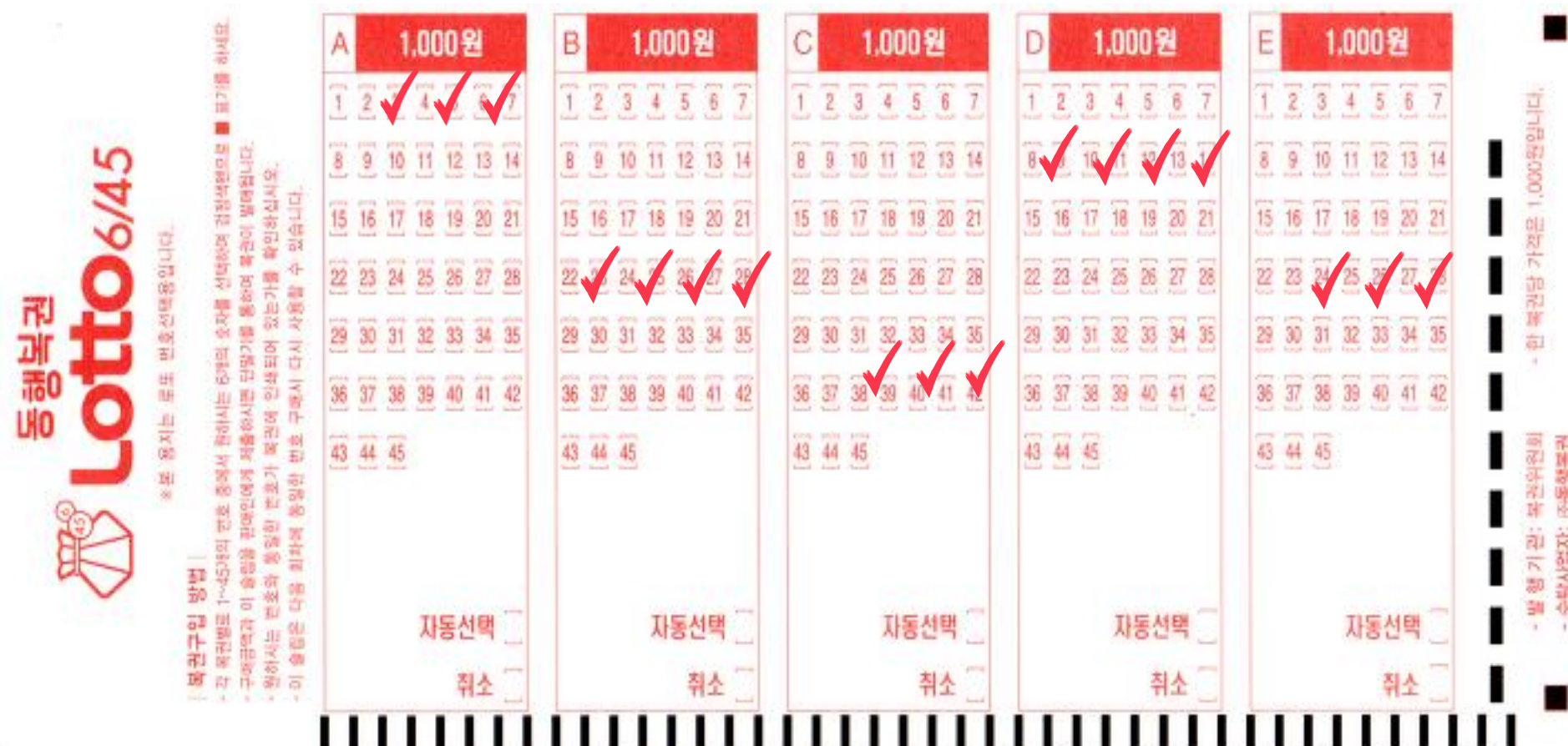
0:6 4%

```
112 while (token != 5)
113 {
114
115     token = evencnt = oddcnt = 0;
116     for (i = 0; i < 3; i++) {
117         if (lotto[i] % 2 == 0)
118             evencnt++;
119         else
120             oddcnt++;
121     }
122
123     for (i = 3; i < 6;) //홀짝비율 맞추기 3:3
124     {
125
126         rannum = rand() % 45;
127
128         if (arr[rannum] != 0 && arr[rannum] != lotto[i - 1] && arr[rannum] != lotto[i - 2]) //중복제거
129         {
130
131             if (arr[rannum] % 2 == 0 && evencnt < 3)
132             {
133                 lotto[i] = arr[rannum];
134                 evencnt++;
135                 i++;
136             }
137             else if (arr[rannum] % 2 == 1 && oddcnt < 3)
138             {
139                 lotto[i] = arr[rannum];
140                 oddcnt++;
141                 i++;
142             }
143         }
144     }
145
146     bubble(lotto, i - 1); //오름차순 정렬
147
148     for (i = 1; i < 6; i++)
149     {
150         if (lotto[i - 1] + 1 != lotto[i] || lotto[i] != lotto[i + 1] - 1) //3번 이상 연속된 수 제거
151             token++;
152     }
153
154 }
```

연속되는
숫자는 피해라!

세 네번

이전 로또 추첨 결과 데이터 분석 후
3연속은 2%, 4연속은 0.1% 라는 통계를 확인했다.

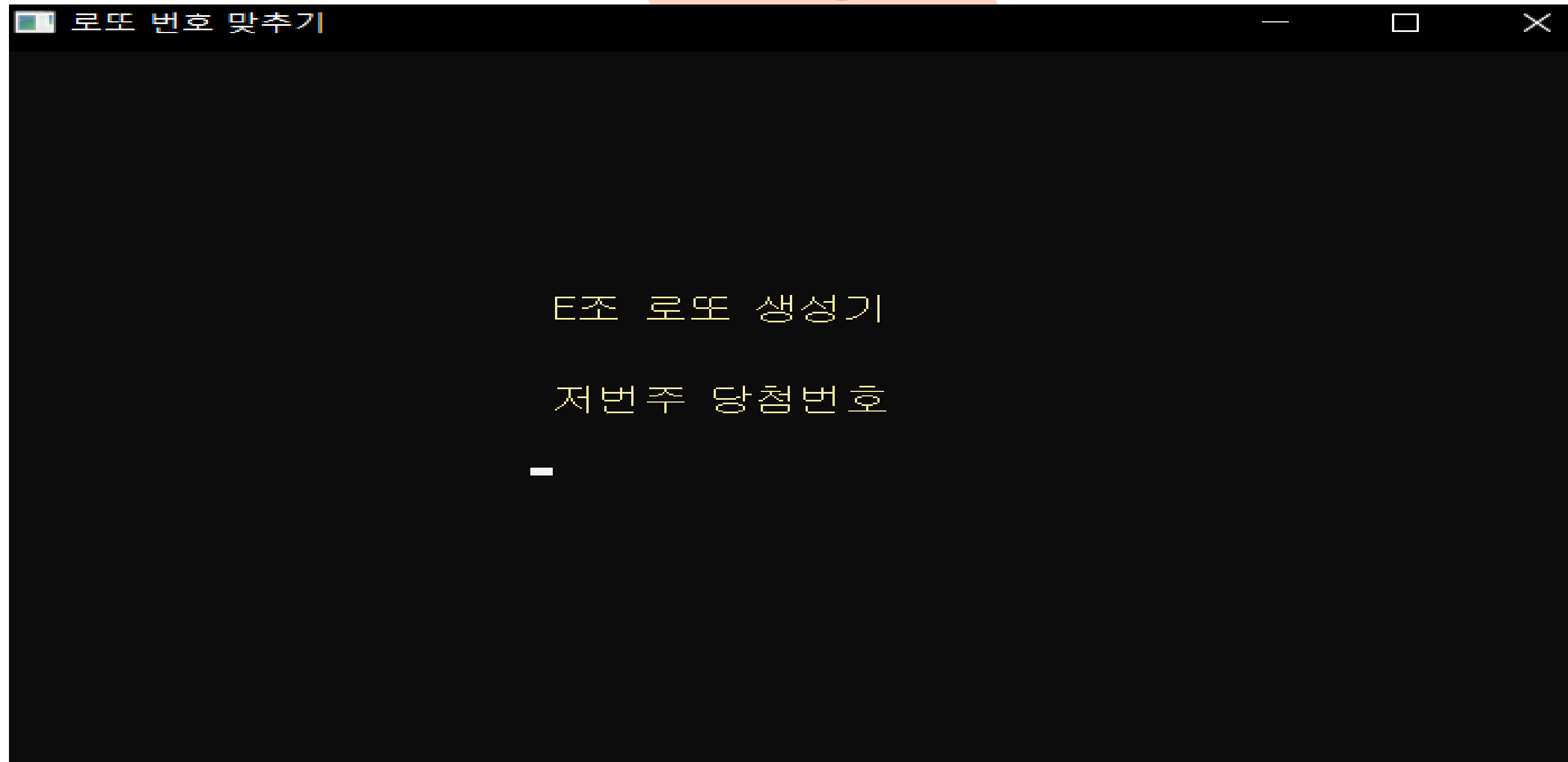


```
20  int bubble(int lotto[], int num) { //오름차순 정렬함수
21      int temp, i, j;
22      for (int i = 0; i < num; i++)
23      {
24          for (int j = 0; j < num; j++)
25          {
26              if (lotto[j] > lotto[j + 1])
27              {
28                  temp = lotto[j];
29                  lotto[j] = lotto[j + 1];
30                  lotto[j + 1] = temp;
31              }
32          }
33      }
34  }

146  bubble(lotto, i - 1); //오름차순 정렬
147
148  for (i = 1; i < 6; i++)
149  {
150      if (lotto[i - 1] + 1 != lotto[i] || lotto[i] != lotto[i + 1] - 1) //3번 이상 연속된 수 제거
151          token++;
152  }
153
154 }
```

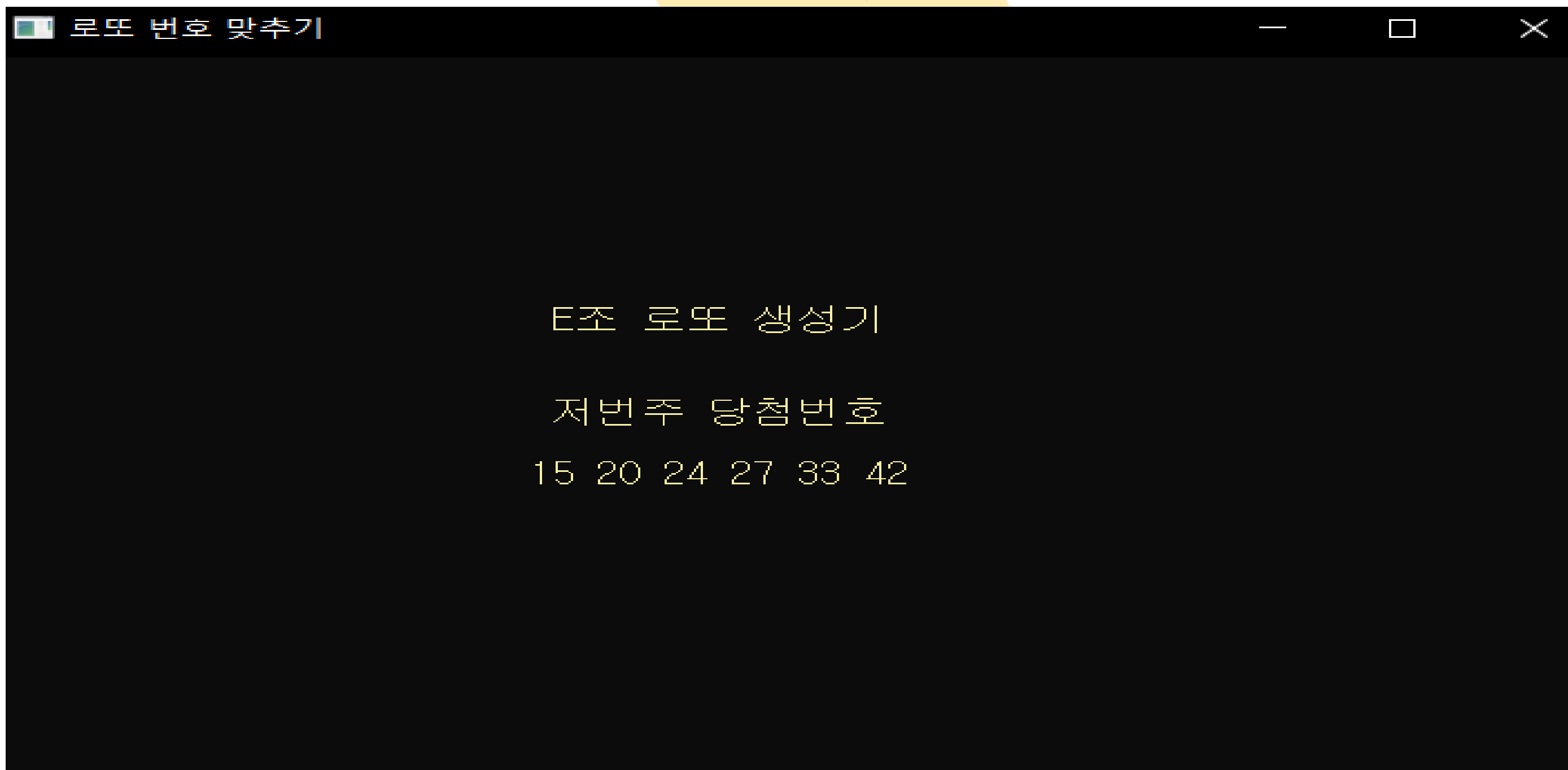
1

실행



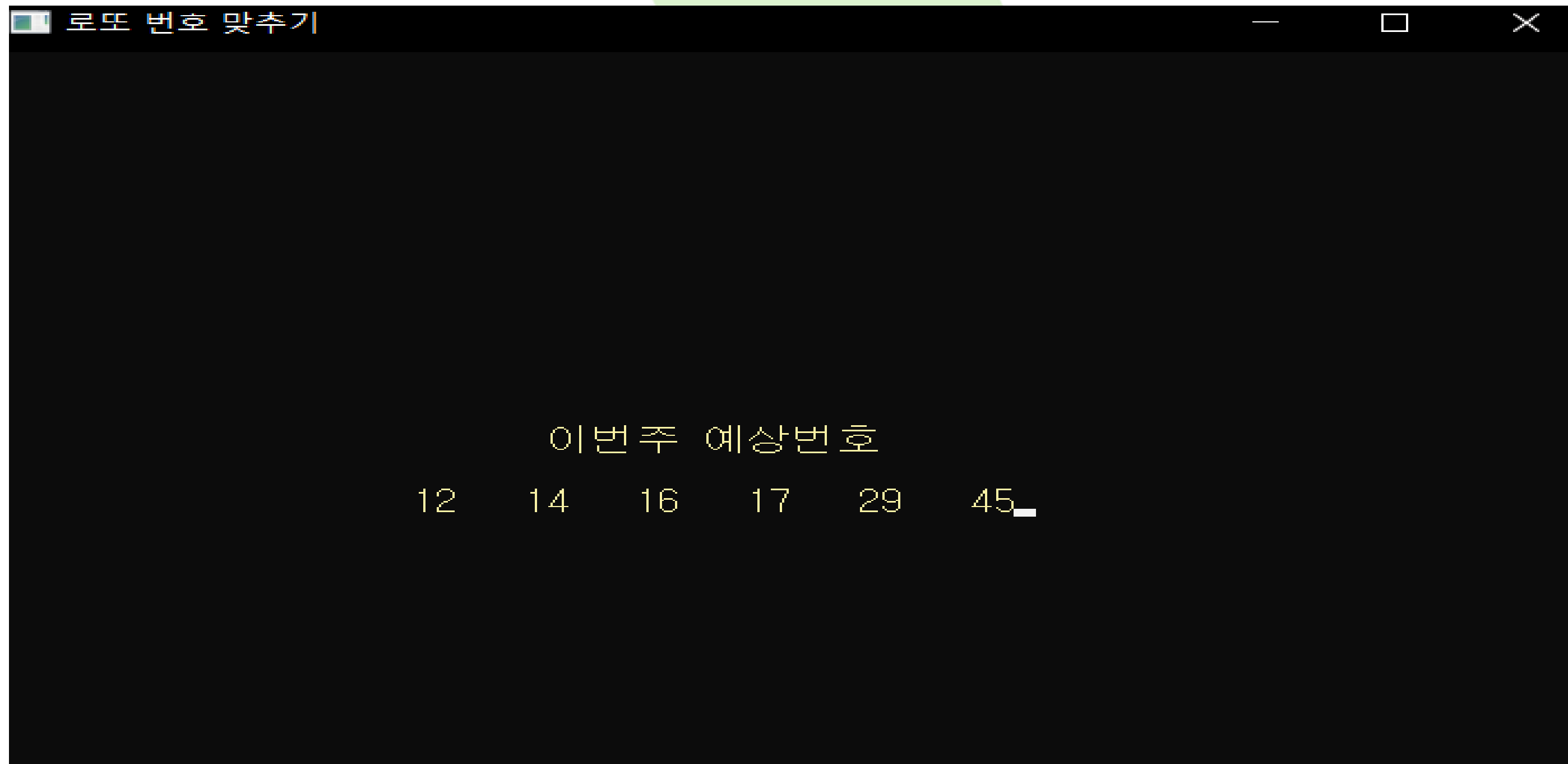
2

입력



3

출력



감사합니다.®

로또 1등 당첨 확률 높이기!

E 조

김 동 욱
김 지 연
이 한 글
최 시 영