


# LOTTO PROGRAM 만들기

곽상우 노진우 허선용 박상우



개요



알고리즘  
선정과정



제작과정



결과

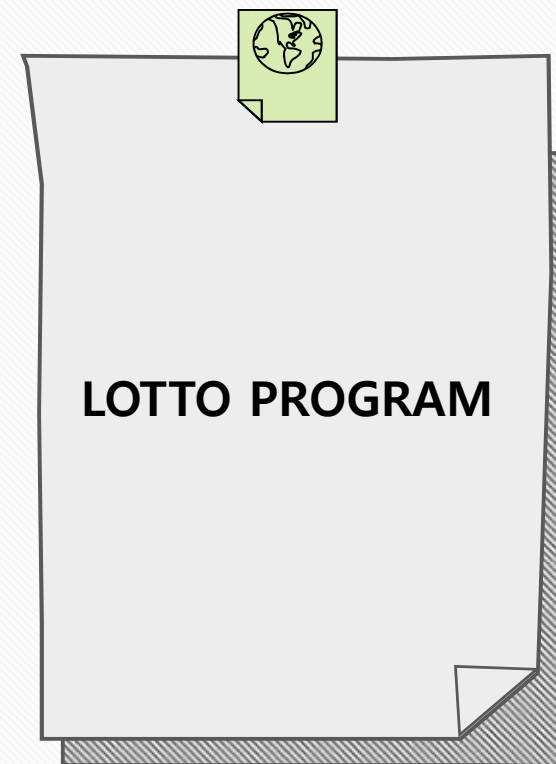


**아이디어 회의 결과**

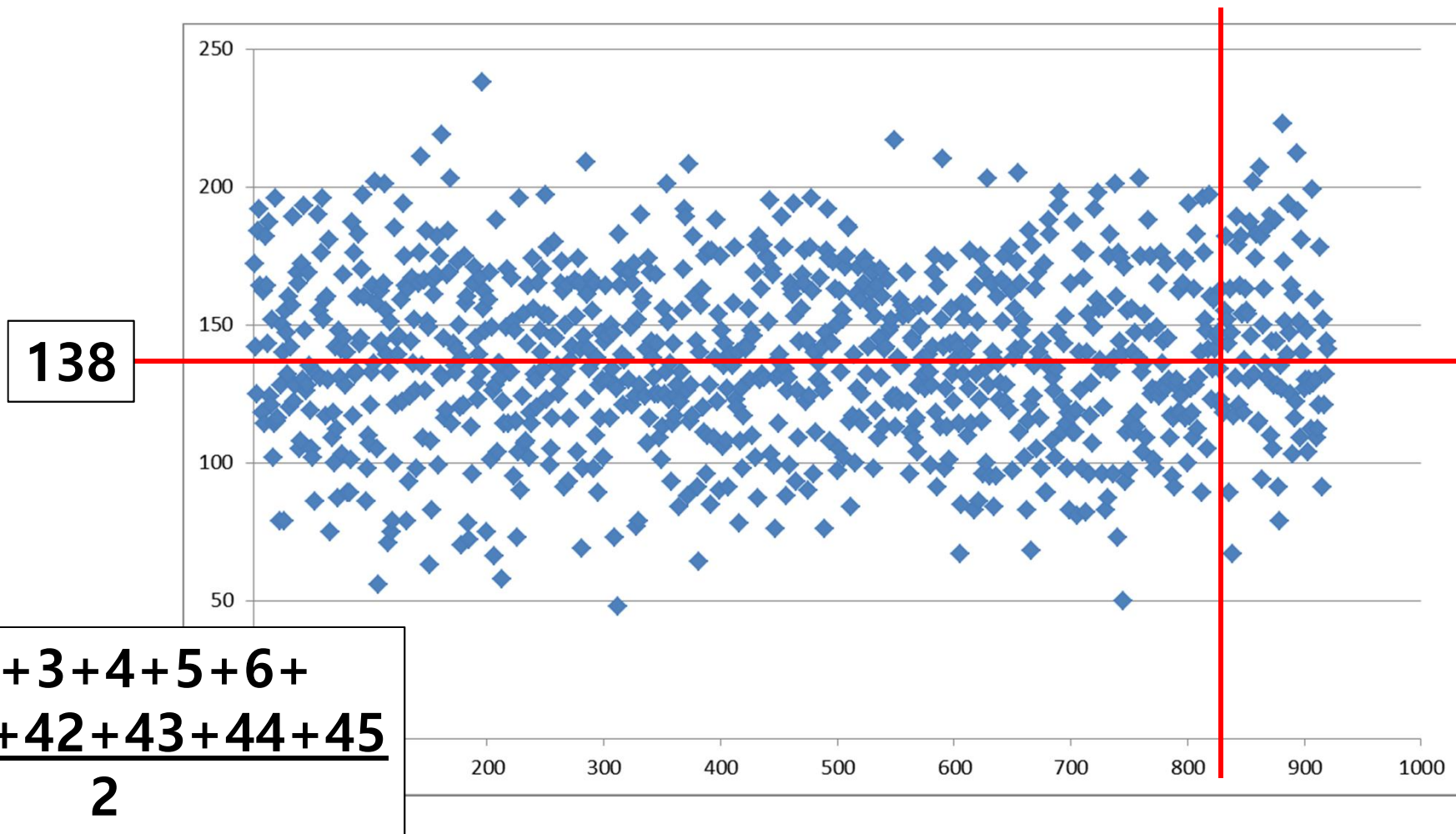
로또배열의 총합  
ex:)1,2,3,4,5,6 -> 21

로또배열의 홀수갯수 분포  
ex:)1,2,3,4,5,6 -> 3

이전 당첨번호의 이월가능성  
(검토 중 보류함)

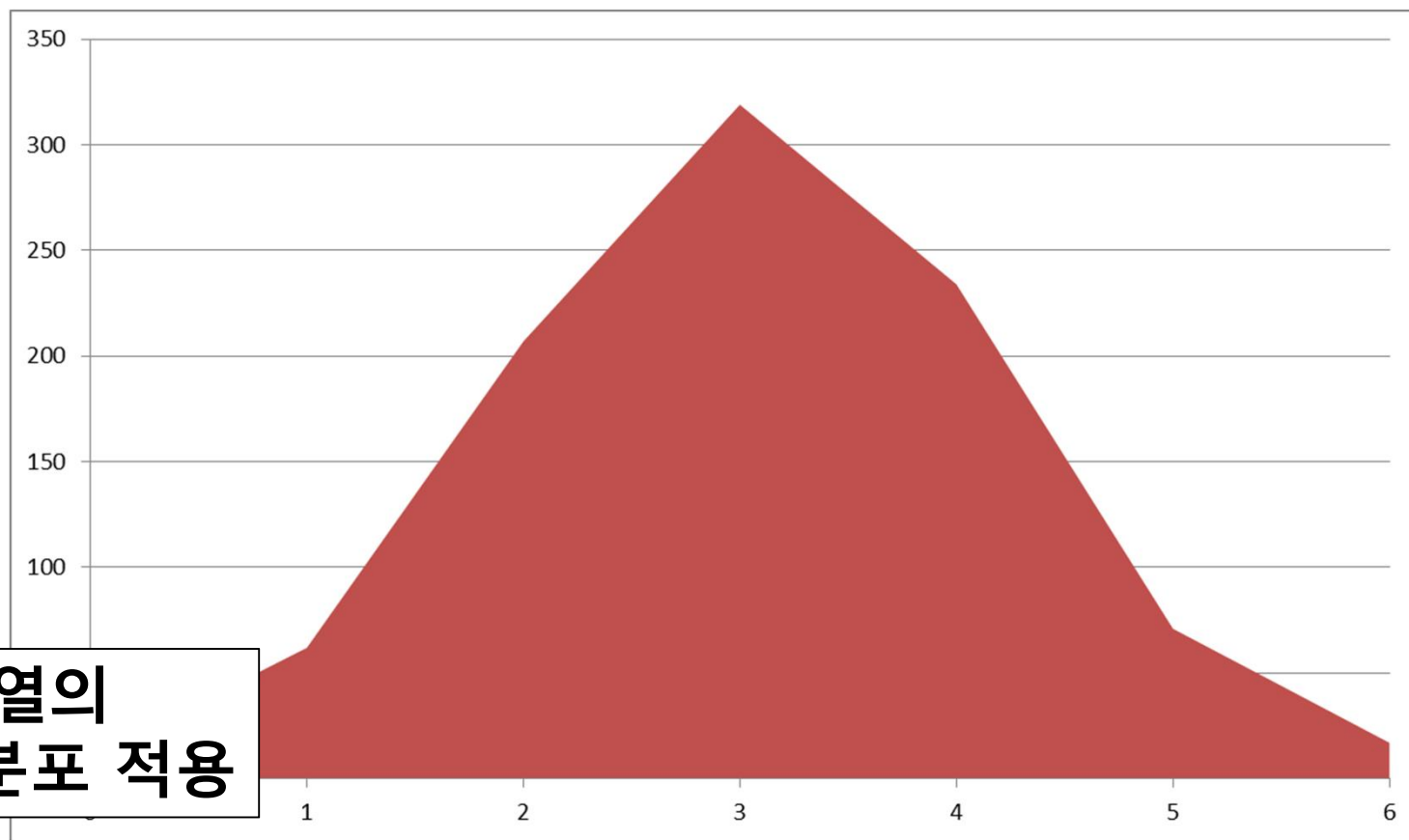


## 로또 배열의 총합



회차

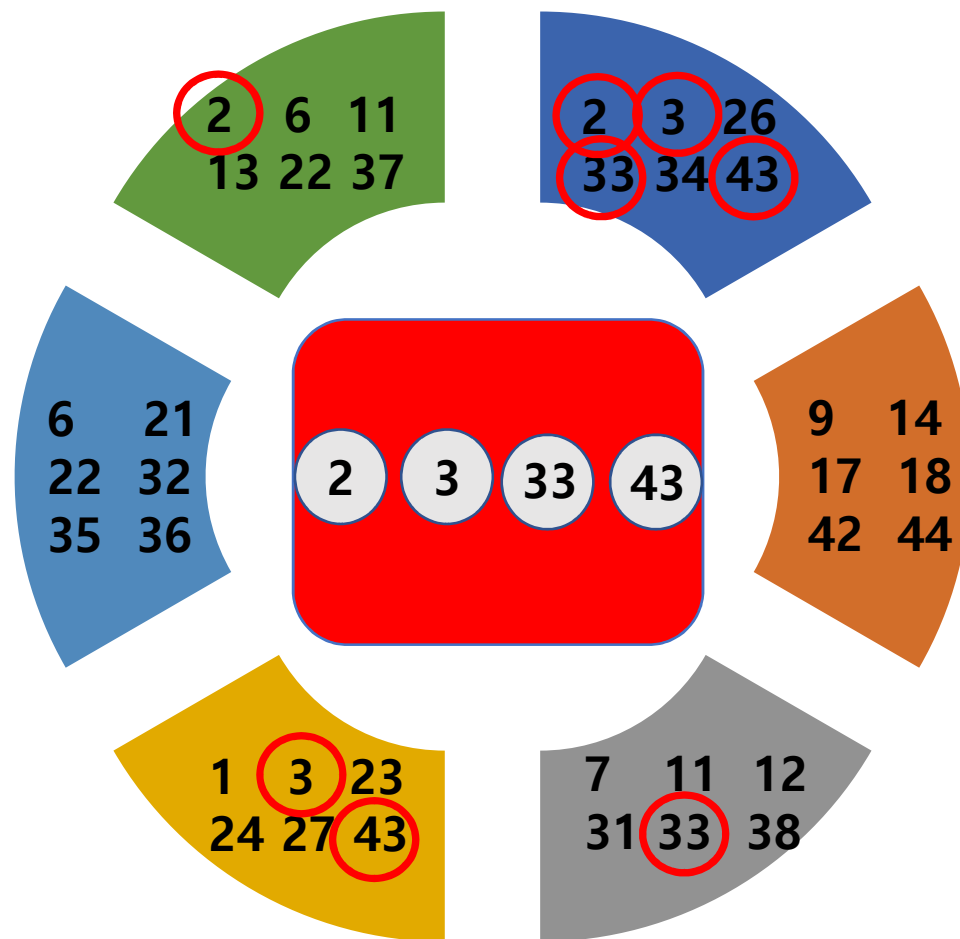
## 로또 배열의 수



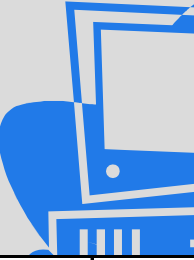
로또 배열의  
홀수 개수 분포 적용

홀수 개수

이  
표



- 920회
- 919회
- 918회
- 917회
- 916회
- 915회



작업	07월 22일	07월 23일	07월 24일	07월 25일
아이디어	1. 로또숫자총합 범위 2. 홀수갯수 분포적용 3. 이전 당첨번호 이월가능성(보류)			
코딩	당첨번호이력 excel -> text -> 참조	조건값산출하기 -> text출력	조건값 참조 -> 결과값 산출	피드백
ppt	ppt작성, 진행척도 확인, 코드 구현 이해			
발표자	ppt발표, 진행척도 확인, 코드 구현 이해			
발표자	ppt발표, 진행척도 확인, 코드 구현 이해			

[illegible]



# 제작과정

## LOTTO PROGRAM

당첨 이력 입력

조건값 출력

결과값 출력

데이터 입력

1. 회차별 당첨번호를 홈페이지에서 다운
2. c언어에서 쉽게 읽을 수 있게 text파일로 수정
3. c언어상에서 읽어들이 데이터 확인

lotto.txt - 메모장					
파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)	
1	10	23	29	33	37
2	9	13	21	25	32
3	11	16	19	21	27
4	14	27	30	31	40
5	16	24	29	40	41
6	14	15	26	27	40
7	2	9	16	25	26
8	8	19	25	34	37
9	2	4	16	17	36
10	9	25	30	33	41
11	1	7	36	37	41
12	2	11	21	25	39
13	22	23	25	37	38
14	2	6	12	31	33
15	3	4	16	30	31
16	6	7	24	37	38
17	3	4	9	17	37
18	3	12	13	19	32
19	6	30	38	39	40
20	10	14	18	20	30
21	6	12	17	18	32
22	4	5	6	8	17
23	5	13	17	18	42
24	7	8	27	29	36
25	2	4	21	26	43
26	4	5	7	18	20
27	1	20	26	28	37
28	9	18	23	25	35
29	1	5	13	34	39
30	8	17	20	35	36
31	7	9	18	23	28
32	6	14	19	25	34
33	4	7	32	33	40
34	9	26	35	37	40
35	2	3	11	26	37
36	1	10	23	26	28
37	7	27	30	33	35
38	16	17	22	30	37
39	6	7	13	15	21
40	7	13	18	19	25
41	13	20	23	35	38
42	17	18	19	21	23
43	6	31	35	38	39
44	3	11	21	30	38
45	1	10	20	27	33
46	8	13	15	23	31
47	14	17	26	31	36
48	6	10	18	26	37
49	4	7	16	19	33
50	2	10	12	15	22

```
896 5 12 25 26 38 45
897 6 7 12 22 26 36
898 18 21 28 35 37 42
899 8 19 20 21 33 39
900 7 13 16 18 35 38
901 5 18 20 23 30 34
902 7 19 23 24 36 39
903 2 15 16 21 22 28
904 2 6 8 26 43 45
905 3 4 16 27 38 40
906 2 5 14 28 31 32
907 21 27 29 38 40 44
908 3 16 21 22 23 44
909 7 24 29 30 34 35
910 1 11 17 27 35 39
911 4 5 12 14 32 42
912 5 8 18 21 22 38
913 6 14 16 21 27 37
914 16 19 24 33 42 44
915 2 6 11 13 22 37
916 6 21 22 32 35 36
917 1 3 23 24 27 43
918 7 11 12 31 33 38
919 9 14 17 18 42 44
920 2 3 26 33 34 43
```

```
int size = file_count();
int** lotto = malloc(sizeof(int*) * size); //이중 포인
for (int i = 0; i < size; i++)
{
    lotto[i] = malloc(sizeof(int) * 7);
}
for (int v = 0; v < size; v++) //입력된 값 초기화
{
    for (int h = 0; h < 7; h++)
    {
        lotto[v][h] = 0;
    }
}

//<조건값을 도출하기> text파일의 내용 입력하기
FILE* text = fopen("lotto.txt", "rt"); //FILE* fopen(
if (text == NULL)
{
    puts("파일오른 실패!"); //puts:문자열만 출력(속도향상)
    return-1; //에러메시지를 반환
}
int temp;
while ((temp = fgetc(text)) != EOF) //text파일의 숫자들
{
    for (int v = 0; v < size; v++)
    {
        for (int h = 0; h < 7; h++)
        {
            fscanf(text, "%d", &temp); //text파일에서
            lotto[v][h] = temp; //text파일에서 받은 회
        }
    }
}
fclose(text); // int fflush(FILE* restrict stream); B
```



# 제작과정

OTTO PROGRAM

당첨 이력 입력

조건값 출력

결과값 출력

```
//<조건1. [로또 번호 6개의 합][최소치와 최대치의 평균 138]을 기준으로 분포를 파악, 비중을 적용>
int n = 0; //최근 당첨회차지정
int spread[3] = { 0, }; //6개의 숫자합이 0:<138 1:=138 2:>138
int temp_sum = 0;
printf("조건1에 사용될 최근 당첨회차 범위를 입력하세요\n");
scanf("%d", &n);
for (int v = 0; v < n; v++)
{
    for (int h = 0; h < 6; h++)
    {
        temp_sum += lotto[size - 1 - v][h + 1]; //h+1:lotto[][0]==
    }
    if (temp_sum < 138)
    {
        spread[0]++;
    }
    else if (temp_sum == 138)
    {
        spread[1]++;
    }
    else
    {
        spread[2]++;
    }
    temp_sum = 0; //다음 최대값, 최소값을 갱신하기 위해 초기화
}
printf("count_down:%d count_same:%d count_up:%d\n", spread[0], spread[1], spread[2]);
//system("PAUSE");
int rand_c1 = (rand() % n) + 1;
int cal_1 = 0;
int c1_cnt = 0;
for (int i = 0; i < 3; i++)
{
    cal_1 += spread[i];
    if (cal_1 < rand_c1)
    {
        c1_cnt++;
    }
}
```

E:\W7\_c\example\Debug\example.exe

```
c1:0 c2:0
조건 생성하기를 선택하셨습니다
조건1에 사용될 최근 당첨회차 범위를 입력하세요
200
count_down:100 count_same:2 count_up:98
rand_c1:73
n:200
6개의 숫자합이 0:<138 1:=138 2:>138 0
10 62 207 319 234 71 17
rand_c2:644
size:920
로또배열의 홀수갯수:4
계속하려면 아무 키나 누르십시오 . . .
```

조건1값 <로또배열의 총합의 분포>

로또배열의 (최대값+최소값)/2 = 138을 기준  
최근범위를 설정하여,  
현재까지 나온 로또배열의 총합들의 분포를  
138보다 높은 경우와 낮은 경우, 같은 경우  
138과 같거나 높ی, 낮게 나올 확률을 적용

# 제작과정

LOTTO PROGRAM

당첨 이력 입력

조건값 출력

결과값 출력

```
////<조건2. 당첨번호중 n개의 홀수가 나온 갯수의 비율을 적용>
int hol_rec[7] = { 0, }; //홀수가 나올 수 있는 경우의 수를 카운트
int temp_hol = 0;
int cnt = 0;
for (int v = 0; v < size; v++)
{
    cnt = 0;
    for (int h = 0; h < 6; h++)
    {
        temp_hol = lotto[size - 1 - v][h + 1];
        if (temp_hol % 2 == 1)
        {
            cnt++; //홀수일 경우 표식
        }
    }
    hol_rec[cnt]++;
}
for (int i = 0; i < 7; i++) //출력값 확인
{
    printf("%d ", hol_rec[i]);
}
printf("\n");
int rand_c2 = (rand() % size) + 1;
int cal_2 = 0;
int c2_cnt = 0;
for (int i = 0; i < 7; i++)
{
    cal_2 += hol_rec[i];
    if (cal_2 < rand_c2)
    {
        c2_cnt++;
    }
}
c2 = c2_cnt; //홀수 갯수
printf("rand_c2:%d\n", rand_c2);
printf("size:%d\n", size);
printf("로또배열의 홀수갯수:%d\n", c2);
```

E:\W7\_c\example\Debug\example.exe

```
c1:0 c2:0
조건 생성하기를 선택하셨습니다

조건1에 사용될 최근 당첨회차 범위를 입력하세요
200
count_down:100 count_same:2 count_up:98
rand_c1:73
n:200
6개의 숫자합이 0:<138 1:=138 2:>138 0
10 62 207 319 234 71 17
rand_c2:644
size:920
로또배열의 홀수갯수:4
계속하려면 아무 키나 누르십시오 . . .
```

조건2값 <로또배열에서 홀수가 n개 나올 분포>

당첨 이력모두를 사용해 홀수가 나온 분포를 확인  
각각의 홀수갯수가 나왔던 확률을 적용

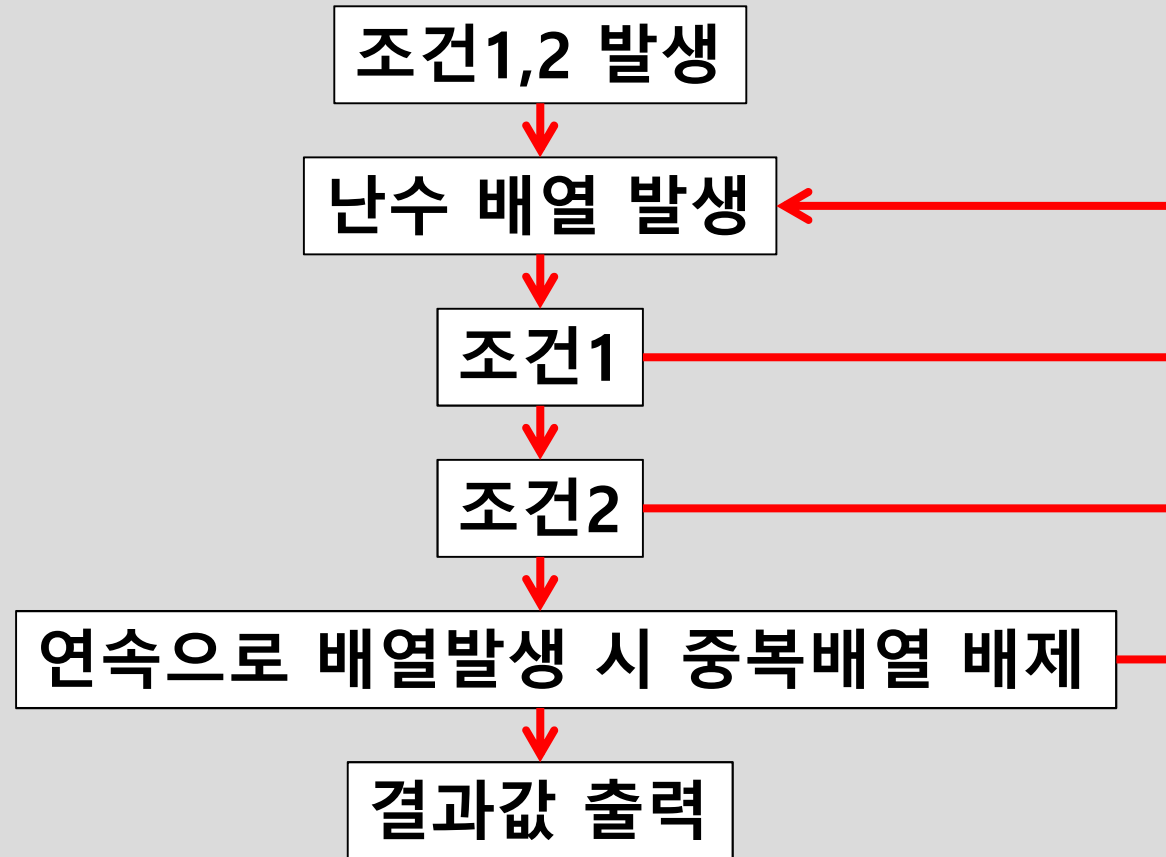
# 제작과정

OTTO PROGRAM

당첨 이력 입력

조건값 출력

결과값 출력



E:\W7\_c\example\WDeb

4	6	12	19	34	42
2	8	18	25	36	40
2	8	15	16	40	42
2	15	20	22	32	36
6	10	12	20	32	33
2	18	20	23	32	36
2	10	16	26	38	41
2	5	10	16	30	34
6	16	18	24	29	32
6	17	20	24	26	30
2	6	9	12	14	34
4	6	8	16	39	44
2	6	8	11	14	32
2	8	10	32	33	34
2	10	16	22	29	30
3	6	10	14	34	38
4	10	12	16	17	24
2	10	12	20	28	43
10	16	18	24	28	29
4	10	12	13	14	40
1	2	12	16	20	24
6	8	14	15	30	34
6	8	10	11	26	38
8	18	22	24	32	33
4	8	22	29	32	40

# 피드백 개선사항

OTTO PROGRAM

1

**END USER**의 요구에 맞게  
조건별 확률 확인 프로그램을  
알고리즘 조건을 명확하게 규정  
유저가 로또배열을 받을 수 있게 수정

2

## 알고리즘

조건으로 제외되던 배열들을  
검출가능하게 변경

3

## ppt개선

알고리즘 설명부를 보기 쉽게 바꿈  
제작과정을 알 수 있게 추가

4

## 코드

쌩다 갈아엎음