



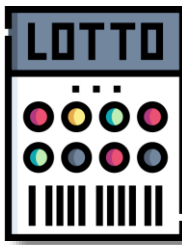
로또 확률 높이기 프로젝트!!



Lotto 6/45



작성자 : 황창식



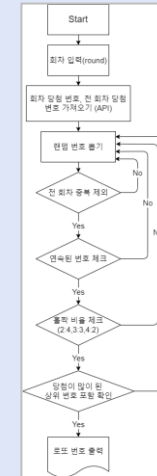
# 목차

1 확률을 높이는 방법

2 순서도 소개

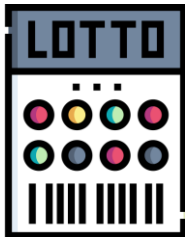
3 코드 소개

4 실제 구동



```
int round = int.Parse(resultRound.Text);
if(round >= 958)
{
    int[] preNums = getNums(957);
    int[] randNums = randNum(preNums);
    int copy = 0;
    for (int i = 0; i < randNums.Length; i++)
    {
        for (int j = i; j < randNums.Length; j++)
        {
            if (randNums[i] > randNums[j])
            {
                copy = randNums[i];
                randNums[i] = randNums[j];
                randNums[j] = copy;
            }
        }
    }
    getResult(randNums);
    getText();
}
else
{
    int[] nowNums = getNums(round);
    int[] preNums = getNums(round - 1);
    int[] randNums = randNum(preNums);
    int copy = 0;
    for (int i = 0; i < randNums.Length; i++)
    {
        for (int j = i; j < randNums.Length; j++)
        {
            if (randNums[i] > randNums[j])
            {
                copy = randNums[i];
                randNums[i] = randNums[j];
                randNums[j] = copy;
            }
        }
    }
}
```





# 1 확률을 높이는 방법

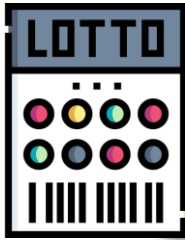
## 1. 지난 회차의 번호를 피하라

지난 회차와 같은 번호가 나올 확률은 3%입니다. 물론 완벽하게 같은 번호가 나올 확률이지만, 45개 중에 6개의 번호를 뽑는 것 보다 39개 중에 6개의 번호를 뽑는게 더욱 당첨확률이 높은게 사실입니다.

## 2. 3번 이상 연속된 숫자를 피하라

역대 1등 번호 중, 3번 이상 연속된 숫자가 당첨된 경우는 292회 딱 한번 있었습니다. 다시 말하면 연속 된 숫자를 제외한다면 확률은 더 올라간다는 말입니다.

1번 연속 : 약 39%  
2번 연속 : 약 11%  
3번 연속 : 약 2%  
4번 연속 : 약 0.1%



# 1 확률을 높이는 방법

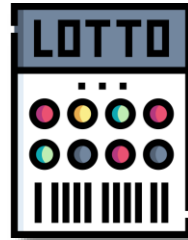
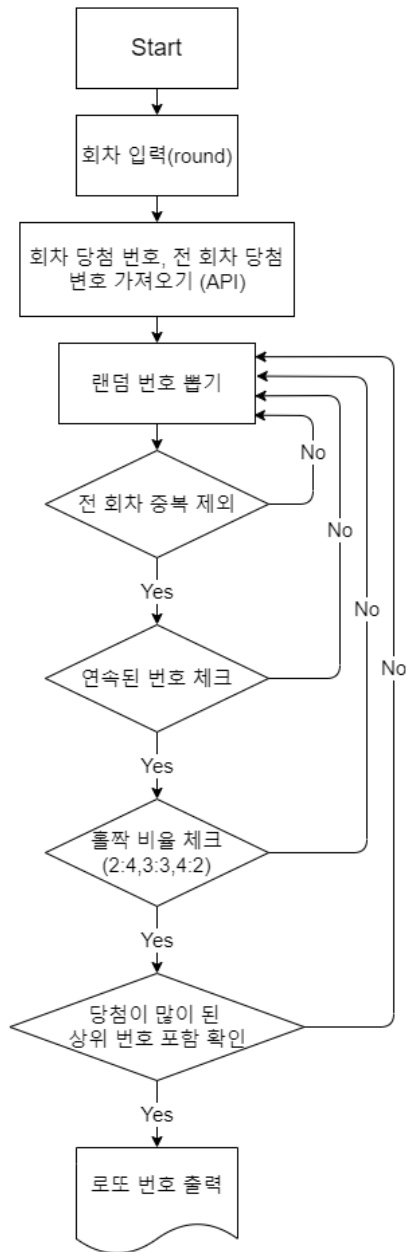
## 3. 홀수와 짝수를 골고루 섞어라

모든 번호가 홀수가 나오거나 짝수가 나온 경우는 전체의 약 3% 밖에 없습니다. 따라서, 홀수와 짝수를 적당히 섞어서 번호를 뽑는 게 확률이 더 높아집니다.

홀수 5회 + 짝수 1회 : 약 7%  
홀수 4회 + 짝수 2회 : 약 22%  
홀수 3회 + 짝수 3회 : 약 34%  
홀수 2회 + 짝수 4회 : 약 26%  
홀수 1회 + 짝수 5회 : 약 7%

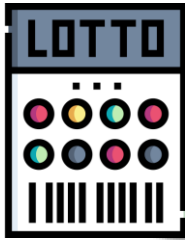
## 4. 가장 많이 나온 번호를 주목하라

최근 1년간의 회차 중 가장 많이 나온 번호(43, 27, 34, 17, 4, 13) 상위 6개를 포함 하는 확률은 약 71%입니다. 따라서 가장 많이 나온 번호 상위 6개를 포함 하는것이 당첨 확률이 더 높습니다.



## 2 순서도 소개

1. 회차 입력한 뒤 해당 회차 당첨번호와 전 회차 당첨 번호를 가져온다.
2. 랜덤번호를 6개 뽑아서 확률 조건들을 체크한다.
3. 조건들 중 하나도 충족하지 않으면, 다시 번호를 뽑는다.
4. 조건들을 모두 충족하면, 번호 출력



## 3 코드 소개

```
int round = int.Parse(resultRound.Text);
if(round >= 958)
{
    int[] preNums = getNums(957);
    int[] randNums = randNum(preNums);
    int copy = 0;
    for (int i = 0; i < randNums.Length; i++)
    {
        for (int j = i; j < randNums.Length; j++)
        {
            if (randNums[i] > randNums[j])
            {
                copy = randNums[i];
                randNums[i] = randNums[j];
                randNums[j] = copy;
            }
        }
    }

    getResult(randNums);

    getText();
}
else
{
    int[] nowNums = getNums(round);

    int[] preNums = getNums(round - 1);

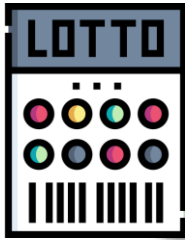
    int[] randNums = sortArr(randNum(preNums));

    getResult(nowNums, randNums);

    getText();
}
```

### 뽑기 버튼 클릭시 코드

1. 입력받은 회차값을 받아와서  
최신 회차 이후 값이면 이전 회차 번호만 불러온다.  
최신 회차 이하 값이면 최신 회차 번호와 이전 회차 번호만 불러온다.
2. 랜덤 값을 불러와서 작은 수부터 정렬한 뒤  
출력하고, 해당 회차 등수와 랜덤 텍스트값을  
출력한다.

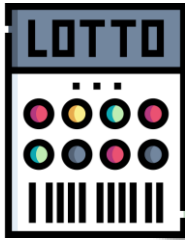


## 3 코드 소개

```
int[] nums = new int[6];
while (true)
{
    for (int i = 0; i < nums.Length; i++)
    {
        nums[i] = r.Next(1, 46);
        for (int j = 0; j < i; j++)
        {
            if (nums[i] == nums[j])
            {
                i--;
                break;
            }
        }
    }
    bool checker = getCheck(preNums, nums);
    if (checker == true)
    {
        checker = getContinue(nums);
        if (checker == true)
        {
            checker = getHoljjack(nums);
            if (checker == true)
            {
                checker = getHignNum(nums);
                if (checker == true)
                {
                    break;
                }
            }
        }
    }
}
return nums;
```

### 랜덤 번호 생성 코드

1. 6 크기의 정수 배열을 생성 한 뒤 랜덤 번호를 하나씩 집어넣는다. 다만, 중복된 번호가 나올 시 다시 번호를 뽑도록 i를 하나 감소시킨다.
2. 확률을 높이는 조건들을 하나씩 체크하면서, 모든 조건이 True가 될 시에 정수 배열을 반환한다. 하나라도 False가 될 시 무한루프를 돌면서 다시 번호를 뽑는다.



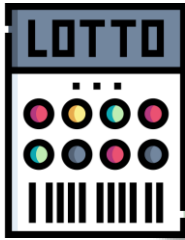
## 3 코드 소개

```
bool checker = true;
for (int i = 0; i < 6; i++)
{
    for (int j = 0; j < 6; j++)
    {
        if (preNums[i] == nums[j])
        {
            checker = false;
            break;
        }
    }
}
return checker;
```

### 전 회차 중복 체크

1. 전 회차 배열과 랜덤으로 뽑은 배열을 하나 하나씩 대조하며 하나라도 일치하는 값이 있으면 False를 반환한다.



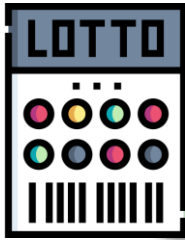


## 3 코드 소개

```
nums = sortArr(nums);  
int count = 0;  
for (int i = 0; i < nums.Length-1; i++)  
{  
    if (nums[i] == nums[i + 1] - 1)  
        count++;  
}  
if (count >= 2)  
    return false;  
else  
    return true;
```

### 연속된 번호 체크

1. 배열을 돌면서 현재 배열값이 다음 배열 값의 -1인 경우를 count한다.
2. count값이 2이상(3개 번호 연속)일 경우 False를 반환하고, 1 이하(2개 번호 연속)일 경우 True를 반환한다.

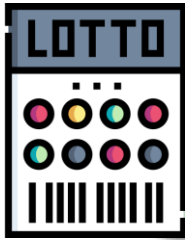


## 3 코드 소개

```
bool checker = false;
int holcount = 0;
int jjackcount = 0;
for (int i = 0; i < nums.Length; i++)
{
    if (nums[i] % 2 == 0)
        jjackcount++;
    else
        holcount++;
}
if (holcount >= 2 || jjackcount >= 2)
    checker = true;
return checker;
```

### 홀짝 비율 체크

1. 배열을 돌면서 홀수의 갯수와 짝수의 갯수를 count한다.
2. 홀수의 count값이 2 이상이거나, 짝수의 count값이 2 이상일 경우에 True를 반환하고, 그 외에는 False를 반환한다.

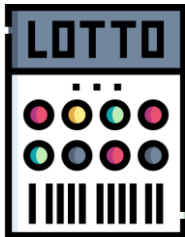


## 3 코드 소개

```
bool checker = false;
for (int i = 0; i < nums.Length; i++)
{
    if(nums[i] == 43 || nums[i] == 27 || nums[i] == 34 || nums[i] == 17 || nums[i] == 4 || nums[i] == 13)
        checker = true;
}
return checker;
```

### 상위 번호 포함 체크

1. 배열을 돌면서 가장 많이 나온 번호 6개의 포함 여부를 체크한다. 하나라도 포함되어 있으면, True를 반환하고, 그 외 경우는 False를 반환한다.



# 3 코드 소개

## API 관련 코드

```
// 로또 API 가져오기
참조 1개
private string GetHttpLottoString(string strUri)
{
    string strResponseText = string.Empty;

    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(strUri);
    request.Method = "GET";

    request.Timeout = 20 * 1000;

    using (HttpWebResponse hwr = (HttpWebResponse)request.GetResponse())
    {
        if (hwr.StatusCode == HttpStatusCode.OK)
        {
            Stream respStream = hwr.GetResponseStream();
            using (StreamReader sr = new StreamReader(respStream))
            {
                strResponseText = sr.ReadToEnd();
            }
        }
        else
        {
            strResponseText = "";
        }
    }
    return strResponseText;
}
```

```
// 회차별 당첨결과 가져오기
참조 3개
public int[] getNums(int round)
{
    string strReturnValue = GetHttpLottoString("https://www.dhlottery.co.kr/" +
        "common.do?method=getLottoNumber&drwno=" + round);
    if (strReturnValue == "")
    {
        MessageBox.Show("불러오기 실패", "오류");
    }

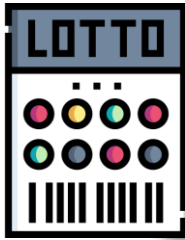
    JsonTextParser jtr = new JsonTextParser();
    JsonObject jobj = jtr.Parse(strReturnValue);

    JsonObjectCollection jac = (JsonObjectCollection)jobj;

    int[] nums = {
        Convert.ToInt32(jac["drwtNo1"].GetValue()),
        Convert.ToInt32(jac["drwtNo2"].GetValue()),
        Convert.ToInt32(jac["drwtNo3"].GetValue()),
        Convert.ToInt32(jac["drwtNo4"].GetValue()),
        Convert.ToInt32(jac["drwtNo5"].GetValue()),
        Convert.ToInt32(jac["drwtNo6"].GetValue()),
        Convert.ToInt32(jac["bnusNo"].GetValue())
    };

    LottoInfo.Text = "누적상금 : " + jac["totSellamnt"].GetValue().ToString() + "원 " +
        "1등 당첨금 : " + jac["firstWinamnt"].GetValue().ToString() + "원 " +
        "1등 당첨 인원 : " + jac["firstPrzwnerCo"].GetValue().ToString() + "명 " +
        "1등 당첨금 총액: " + jac["firstAccumamnt"].GetValue().ToString() + "원";

    return nums;
}
```



# 4 프로그램 구동

로또 번호 프로그램

Created by Yorgos

회차 입력        +

행운 뽑기

"돈을 빌려준 사람은 돈을 빌린  
사람보다 훨씬 기억력이 좋다."

- 프랭클린 -

회차 입력        +  **광!!**

누적상금 : 48769851000원 1등 당첨금 : 1635711483원 1등 당첨 인원 : 7명 1등 당첨금 총액: 11449980375원

행운 뽑기

감사합니다~~!

