

C언어를 활용한 연금복권 자동 번호 생성 코드 만들기

김지환 민경환 박재형 이경희 이상룡

다가오는 100세 시대
나는 연금복권으로 준비한다!

INDEX

1

프로젝트 개요

2

프로젝트 수행 관련이론

3

프로젝트 수행 절차 및 방법

4

프로젝트 수행 결과

프로젝트개요

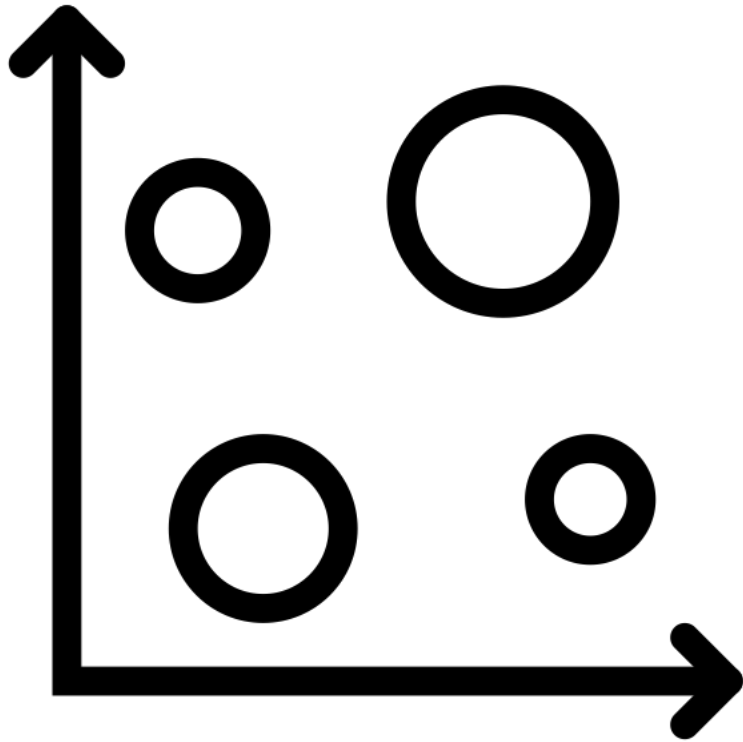


1부터 5까지 5개로 구성된 조단위와

십만자리부터 일자리까지 0에서 9로 구성된 연금복권

지금까지 배운 C언어를 바탕으로 한 연금복권 자동 생성 프로그램을 만들어

조건과 확률에 따라 당첨에 근접한 번호를 추천할 수 있다.



// 상관관계분석 //

상관관계분석이란?

변수간의 밀접한 정도 즉 상관관계를 분석하는 통계적 분석방법으로 양적변수인 두 변수의 선형관계에 대한 유의성을 검증한다.

1. 유의성 검정

두 변수의 선형 관계의 유의성을 검정한다.

2. 크기 비교

상관계수(r)의 크기를 비교하여 더 큰 관계를 파악.



" 시계열분석 "

시계열분석이란?

시계열 데이터를 바탕으로 둔 분석방법으로
동일한 현상을 시간의 경과를 일정 간격을 두고
반복 측정을 통하여 변화 분석하는 방법.

.....

같은 시간 간격으로 순서적으로 배열된 관측값으로
반드시 시간 순서에 의해 관측값을 분석해야 한다.

프로젝트 수행절차

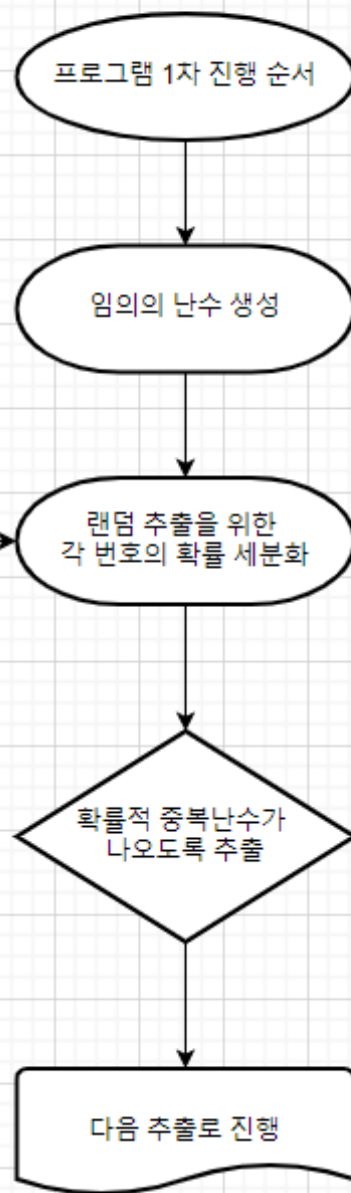


프로젝트 수행절차



```
if (num[k] <= 11)
{
    num[k] = 1;
    printf("%d ", num[k]);
}
else if (num[k] <= 21 && num[k] > 11) {
    num[k] = 2;
    printf("%d ", num[k]);
}
else if (num[k] <= 32 && num[k] > 21) {
    num[k] = 3;
    printf("%d ", num[k]);
}
else if (num[k] <= 40 && num[k] > 32) {
    num[k] = 4;
    printf("%d ", num[k]);
}
else if (num[k] <= 52 && num[k] > 40) {
    num[k] = 5;
    printf("%d ", num[k]);
}
else if (num[k] <= 61 && num[k] > 52) {
    num[k] = 6;
    printf("%d ", num[k]);
}
else if (num[k] <= 72 && num[k] > 61) {
    num[k] = 7;
    printf("%d ", num[k]);
}
else if (num[k] <= 81 && num[k] > 72) {
    num[k] = 8;
    printf("%d ", num[k]);
}
else if (num[k] <= 91 && num[k] > 81) {
    num[k] = 9;
    printf("%d ", num[k]);
}
```

추첨 번호는 각자 다른
확률로 생성



회차별
중복체크



<85%로 중복난수가 나오는 추출식>

```
for (i = 0; k > i; i++)
    if (num[k] == num[i])
    {
        arr[j] = rand() % 100; //0-99 임의의 난수 추출
        if (arr[j] < 85) // rand결과값 num[i] 가 0-99 중에 85이하이면 그대로 추출
        {
```

회차별 중복체크



68	V	5조 191075	191075	91075	1075	075	75	5	218776
67		3조 807531	807531	07531	7531	531	31	1	137560
66	V	4조 990834	990834	90834	0834	834	34	4	364914
65	V	4조 732375	732375	32375	2375	375	75	5	020321
64	V	5조 676003	676003	76003	6003	003	03	3	855524
63	V	4조 335226	335226	35226	5226	226	26	6	791181
62	V	4조 014073	014073	14073	4073	073	73	3	572879
61	V	3조 575862	575862	75862	5862	862	62	2	602610
60	V	5조 216510	216510	16510	6510	510	10	0	029163
59	V	2조 159056	159056	59056	9056	056	56	6	973521
58	V	5조 939327	939327	39327	9327	327	27	7	551924
57	V	2조 272286	272286	72286	2286	286	86	6	230295
56	V	1조 993174	993174	93174	3174	174	74	4	677799
55	V	1조 477879	477879	77879	7879	879	79	9	495719
54		5조 627054	627054	27054	7054	054	54	4	932535
53	V	5조 649638	649638	49638	9638	638	38	8	580357

52	V	2조 597227	597227	97227	7227	227	27	7	297625
51	V	2조 510012	510012	10012	0012	012	12	2	088185
50	V	3조 592111	592111	92111	2111	111	11	1	176570
49		4조 305248	305248	05248	5248	248	48	8	191235
48	V	1조 779363	779363	79363	9363	363	63	3	788642
47		4조 941327	941327	41327	1327	327	27	7	524495
46	V	4조 010963	010963	10963	0963	963	63	3	710817
45		2조 318754	318754	18754	8754	754	54	4	494221
44	V	3조 668818	668818	68818	8818	818	18	8	799181
43	V	4조 589839	589839	89839	9839	839	39	9	890546
42		4조 974231	974231	74231	4231	231	31	1	892243
41	V	1조 742714	742714	42714	2714	714	14	4	698335
40		2조 631086	631086	31086	1086	086	86	6	585802
39	V	1조 199137	199137	99137	9137	137	37	7	320718
38	V	3조 307836	307836	07836	7836	836	36	6	162480
37	V	3조 453877	453877	53877	3877	877	77	7	466566
36	V	4조 735757	735757	35757	5757	757	57	7	252663

프로젝트 수행절차



<난수 선별을 위한 코드>

```
if (num[k] < 10)
{
    if (k == 2)
    {
        if (num[k] == 2 || num[k] == 4 || num[k] == 7 || num[k] == 9)
        {
            num[k];
        }
    }
    else if (k == 3)
    {
        if (num[k] == 0 || num[k] == 2 || num[k] == 6 || num[k] == 8)
        {
            num[k];
        }
    }
    else if (i == 4)
    {
        if (num[k] == 1 || num[k] == 3 || num[k] == 5 || num[k] == 7)
        {
            num[k];
        }
    }
    else
    {
        num[k];
    }
}
```

1차 진행으로 추출된 난수

확률 적용을 통한 난수
한정 방지

특정 확률을 높이는 값이
해당하게 되도록 선별

추첨 숫자 카운팅 후
사용자에게 추천

결과 출력

금주의 연금복권 당첨 예상번호

4조	9	7	7	0	5	4
4조	4	2	5	6	3	8
3조	1	1	1	4	5	3
3조	9	7	3	3	5	4
1조	8	2	7	7	6	0

*****0의	개	수	:	2*****
*****1의	개	수	:	3*****
*****2의	개	수	:	2*****
*****3의	개	수	:	4*****
*****4의	개	수	:	4*****
*****5의	개	수	:	4*****
*****6의	개	수	:	2*****
*****7의	개	수	:	5*****
*****8의	개	수	:	2*****
*****9의	개	수	:	2*****

5번 나온 숫자가 가장 추천될 확률이 높은 숫자입니다.

출력 결과 예시

프로젝트 수행절차



01

생성 100개의 난수를 랜덤에 의해 85% 확률로 당첨 번호 조합을 추출

02

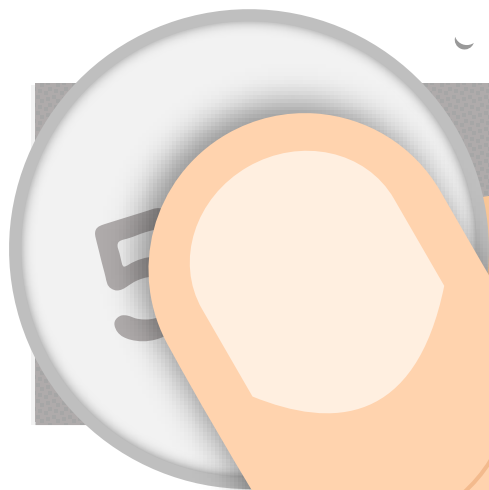
자료 구체화 연속 번호 출현, 번호 별 출현 빈도수에 대한 확률 수치화 (점수화)

03

결과 분석 프로그램 실행으로 산출된 예상 번호와 번호별 출력빈도 비교

04

추가 정보 제공 번호별 출력빈도를 비교하여 추천될 확률이 높은 숫자 제공



작침금 5조

Code

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int num[6];
int arr[1];
int group;
int i = 0, j = 0, k = 0;
int a = 0, b = 0, c = 0, d = 0, e = 0;
int f = 0, g = 0, x = 0, y = 0, z = 0;
int cnt = 0, q;;
srand((unsigned)time(NULL));

printf("*금주의 연금복권 당첨 예상번호*\n\n");
```

Code

```
for (q = 0; q < 5; q++)  
{  
  
    group = rand() % 5 + 1;  
    printf("%4d조 ", group);  
  
    for (k = 0; k < 6; ++k)  
    {  
        num[k] = rand() % 100 + 1; //  
  
        if (num[k] <= 10)  
        {  
            num[k] = 1;  
  
        }  
    }  
}
```

반복문 & 조건문

Code

```
else if (num[k] <= 20 && num[k] > 10)
{
    num[k] = 2;

}
else if (num[k] <= 30 && num[k] > 20)
{
    num[k] = 3;

}
else if (num[k] <= 40 && num[k] > 30)
{
    num[k] = 4;

}
else if (num[k] <= 50 && num[k] > 40)
{
    num[k] = 5;

}
```

Code

```
else if (num[k] <= 60 && num[k] > 50)
{
    num[k] = 6;
}
else if (num[k] <= 70 && num[k] > 60)
{
    num[k] = 7;
}
else if (num[k] <= 80 && num[k] > 70)
{
    num[k] = 8;
}
else if (num[k] <= 90 && num[k] > 80)
{
    num[k] = 9;
}
else if (num[k] <= 100 && num[k] > 90)
{
    num[k] = 0;
}
```

Code

0부터 99까지 임의의 난수를 추출하여
Rand 결과값 num[i]가 85이하일 경우
그대로 추출한다. ➔

```
for (i = 0; k > i; i++)
if (num[k] == num[i])
{
arr[j] = rand() % 100;
if (arr[j] < 85)
{
if (num[k] < 10)
{
if (k == 2)
{
if (num[k] == 2 || num[k] == 4 || num[k] == 7 ||
num[k] == 9)
{
num[k];
}
}
}
```


Code

```
else if (k == 3)
{
    if (num[k] == 0 || num[k] == 2 || num[k] == 6 || num[k] == 8)
    {
        num[k];
    }
}
else if (i == 4)
{
    if (num[k] == 1 || num[k] == 3 || num[k] == 5 || num[k] == 7)
    {
        num[k];
    }
}
else
{
    num[k];
}
}
else if (arr[j] >= 85)
{
    num[k] = rand() % 10;
}
break;
}
printf("%d ", num[k]);
```

Code

IF문을 사용하여 0일때 a에 나온 횟수를 누적하고



IF를 제외한 1부터 9까지의 숫자에 관하여
ELSE IF문을 사용하여 a~z에 나온 횟수를 누적한다.

```
if (num[k] == 0)
    a++;
else if (num[k] == 1)
    b++;
else if (num[k] == 2)
    c++;
else if (num[k] == 3)
    d++;
else if (num[k] == 4)
    e++;
else if (num[k] == 5)
    f++;
else if (num[k] == 6)
    g++;
else if (num[k] == 7)
    x++;
else if (num[k] == 8)
    y++;
else if (num[k] == 9)
    z++;
}
printf("\n");
}
```

Code

번호별 누적값을
배열에 넣어
최댓값을 출력한다. →
temp[1]부터 비교하기 위한 반복문으로
0~9까지의 개수를 출력한다. →
temp[0]이 저장된 max를
temp[1]부터 저장된 값과 비교하여
temp[cnt]가 max보다 크면 max에 대입한다. →

출력

```
printf("\n\n");
printf("*****0의 갯수 : %d*****\n", a);
printf("*****1의 갯수 : %d*****\n", b);
printf("*****2의 갯수 : %d*****\n", c);
printf("*****3의 갯수 : %d*****\n", d);
printf("*****4의 갯수 : %d*****\n", e);
printf("*****5의 갯수 : %d*****\n", f);
printf("*****6의 갯수 : %d*****\n", g);
printf("*****7의 갯수 : %d*****\n", x);
printf("*****8의 갯수 : %d*****\n", y);
printf("*****9의 갯수 : %d*****\n", z);
int temp[10] = { a,b,c,d,e,f,g,x,y,z };
int max = temp[0];
for (cnt = 1; cnt < 10; cnt++) {
    if (max < temp[cnt]) {max = temp[cnt];
    }
}
printf("\n");
printf("%d번 나온 숫자가 가장 추천될 확률이 높은 숫자입니다.\n",
max);
return 0;
}
```

프로젝트 수행결과



```
Microsoft Visual Studio 디버그 콘솔
*금주의 연금복권 당첨 예상번호*

2조 1 7 5 6 6 4
4조 8 0 5 0 0 6
1조 0 2 9 8 8 6
2조 6 2 3 0 0 1
1조 8 7 7 4 4 6

*****0의 갯수 : 6*****
*****1의 갯수 : 2*****
*****2의 갯수 : 2*****
*****3의 갯수 : 1*****
*****4의 갯수 : 3*****
*****5의 갯수 : 2*****
*****6의 갯수 : 6*****
*****7의 갯수 : 3*****
*****8의 갯수 : 4*****
*****9의 갯수 : 1*****

6번 나온 숫자가 가장 추첨될 확률이 높은 숫자입니다.

C:\Users\leekyeonghui\source\repos\Project1\Debug\Project1.exe(프로세스 5940개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

감사합니다.