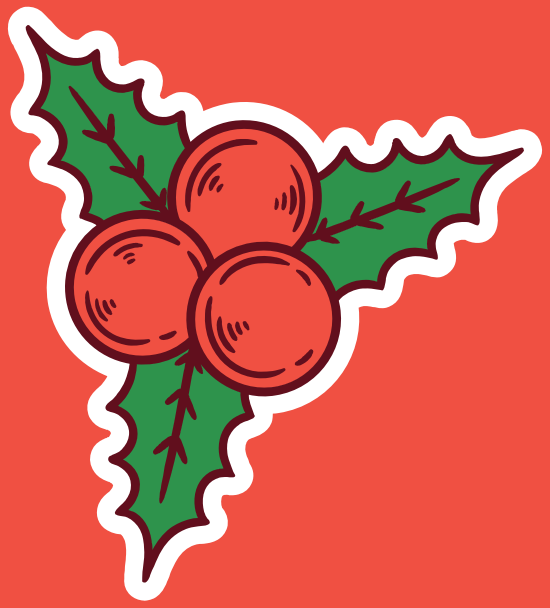


C 언어 스터디

전응진, 최우철, 허지혜

목 차

01

선 택 문

선 택 문

If 문 형식

구분 형식	
if (조건 식) 실행 문;	조건식이 참일 때 실행문 실행 거짓이면 아무것도 실행하지 않음

if (조건식)
실행문1;
else
실행문2;
	조건식이 참이면 실행문
	1 실행 조건식이
	거짓이면 실행문 2
	실행

if (조건식1)
실행문
1; else if
(조건식2)
실행문2;
else
실행문3;
	조건식1이 참이면 실행문 1 실행
	조건식1이 거짓이고 조건식2가 참이면
	실행문 2 실행모든 조건식이 거짓이면
	실행문 3 실행

Switch case : 범위를 보다 한정적으로 지정
조건은 정수식만 사용
Case는 Break를 포함

예를 들면,

Switch(score)

Case 100 : printf('A입니다')

Case 90 : printf('B 입니다')

Default : printf(' 공부하십쇼')

02

반 복 문

반 복 문

반복문은 여러 종류가 있지만, 참고서적에서 while, for, do~while문을 다룸

1. While : 조건을 만족한다면, 실행

2. For : 조건식을 주어, 해당하는 횟수만큼 반복

3. Do~ while : 우선 실행한 후 , 해당 조건을 만족한다면, 순환

반복문 형식	실행 방식

while (조건식)
{
실행문;
}
	조건식이 참인 동안 실행문을 반복한다.
	최초 조건식이 거짓이면 실행문은 한 번도 실행되지 않는다.

for (초기식; 조건식; 증감식)
{
실행문;
}
	초기식은 최초 한 번 실행한다.
	조건식을 검사하여 참이면
	실행문 -> 증감식 -> 조건식을 반복한다.

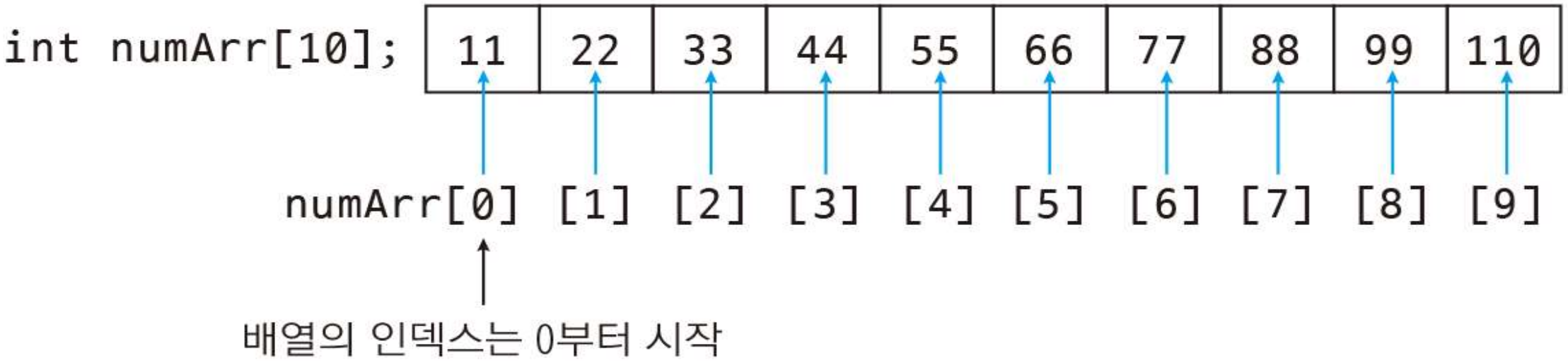
do { 실행문; } while (조건식);	실행문을 수행한 후에 조건을 검사한다. 조건식이 참인 동안 실행문을 반복한다. 실행문은 조건식과 관계없이 최소 한 번은 실행된다.

03

배 열

배 열

- 배열을 선언하면 많은 변수를 한 번에 선언하는 효과를 가진다.
- 배열 초기화 시 중괄호를 사용한다.
- 배열 전체의 크기는 sizeof 연산자를 이용한다.



04

포 인 터

"포인터 = 주소"

주소는 변수가 할당된
메모리 공간의 시작주소.
&변수명 으로 표기한다.

포인터를 쓰는 이유?

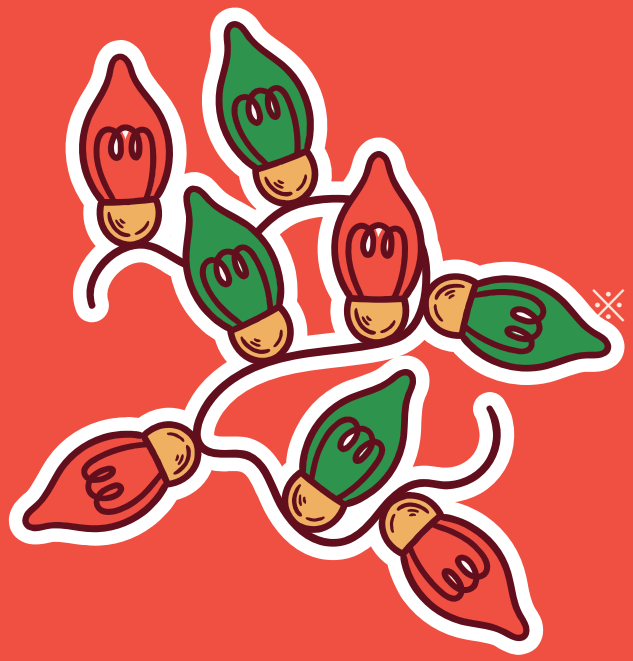
- 함수 간에 효과적으로 데이터를 공유

- 임베디드 프로그래밍시 메모리에 직접 접근하거나

동적 할당한 메모리 사용 시 포인터를 사용

※ 임베디드 프로그래밍: 특정 기능을 제어하는 프로그램

ex) 정수/냉수 구분하여 물이 나오게 하는 것



선언방법

일반변수

Int a; 로 선언한다.

포인터

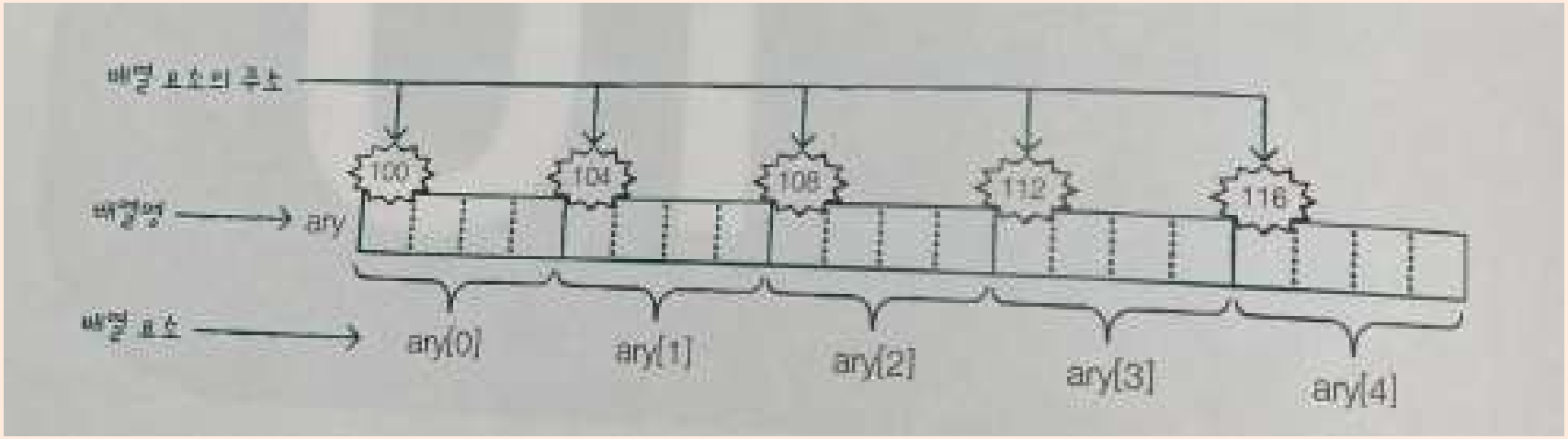
앞에 *를 붙인다.
ex) int *pa;

주의할 점

Const, 즉 상수로 포인터 변수를 선언하면 변수 내부의 값을 변경할 수 없다!

배열과 포인터의 관계

- 배열은 자료형이 갖는 변수를 메모리에 연속으로 할당
- 각 배열 요소는 일정 간격으로 주소를 갖는데, 첫번째 요소의 주소를 알면 나머지 요소 주소도 쉽게 알 수 있으며, 포인터 연산으로 모든 배열 요소를 사용할 수 있다.


배열 요소의 주소
100
104
108
112
116
배열명
ary
배열 요소
ary[0]
ary[1]
ary[2]
ary[3]
ary[4]

배열과 포인터의 차이

배열

- sizeof(배열명) : 배열 전체 크기
- 배열명은 값 변경 불가

포인터

- sizeof(포인터) : 포인터 하나의 크기
- 포인터는 값 변경 가능

05

실 습

강화게임

1. Cmd 창에 나타날 화면

```c
#include <stdio.h>
#include <time.h>
#include <windows.h>

void printG(int money,int lev) //현재 보유중인 골드와 레벨을 표시해주는 함수.
{
    printf(" ┌──────────────────┐ \n");
    printf("  %d G   | Level : %d \n",money,lev);
    printf(" └──────────────────┘ \n");
}
```

2. Upgrade 함수 정의

```c
int upgrade(int level,int luck,int*ptr)
{
    int i;
    srand((unsigned int)time(NULL)); //현재 시각의 값을 rand의 seed값으로 적용.
    int random=rand()%100+5; //5~100까지의 난수를 생성.
    printf("강화중");
    for(i=0; i<3; i++)
    {
        printf(".");
        Sleep(1000); //3초 대기..
    }
    printf("\n뽑힌 랜덤수: %d\n",random);
    if(random<luck)
    {
        printf("성공! +80 G\n");
        *ptr= *ptr+80; //메인함수에 있는 money 변수에 +80
        return level+1; //강화성공에 따라 레벨 1 증가.
    }
    else
    {
        printf("실패..\n");
        return level; //실패시 기존 레벨을 반환.
    }
}
```

강화게임

3. Main 정의

```
int main()
{
    char ch;
    int maxluck, money=200, level=1;
    int cost=level*50;
    int *ptr=&money;
    printf("|######<강화 시뮬레이션 프로그램입니다>######|₩n");
    printf("|@현재 컴퓨터 시간에 따라 랜덤숫자가 뽑히는데|₩n|@랜덤수가 강화 성
    while(1)
    {
        printG(money,level);
        maxluck=100-level*5; //레벨이 1 증가함에 따라 강화확률 -5%
        printf("┌──────┬──────┐₩n");
        printf("|비용:%d G |%d%% 확률| 강화하시겠습니까? Y/N₩n",cost,maxluck);
        printf("└──────┴──────┘₩n");
        ch=getch();
        system("cls"); //화면의 모든것을 지운다.
```

강화게임

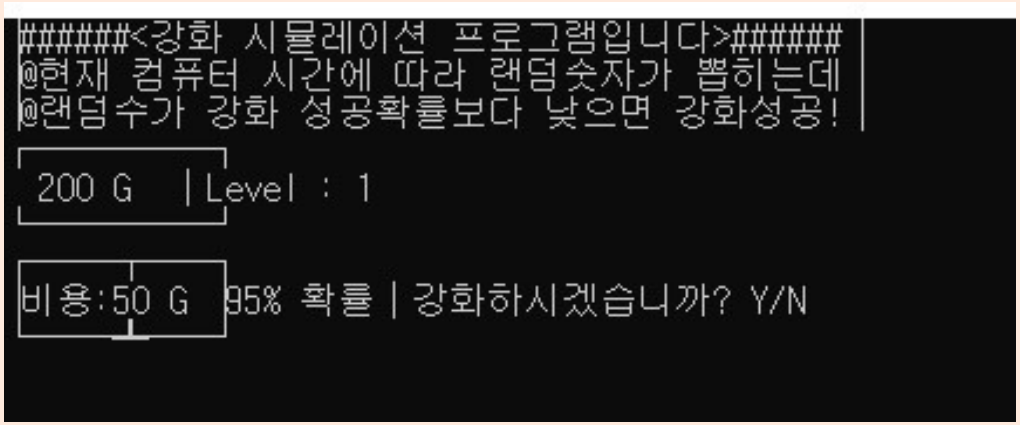
4. if문으로 게임 종료

```
if(ch=='y'|| ch=='Y')
{
    if(money-cost>=0)
    {
        money=money-cost;
        level=upgrade(level,maxluck,ptr);
    }
    else
    {
        printf("골드가 부족합니다₩n");
        break; //골드가 부족할 경우 게임오버
    }
}
```

```
else if(ch=='n'|| ch=='N')
{
    printf("종료하시겠습니까? Y/N₩n");
    ch=getch();
    if(ch=='y'|| ch=='Y')
    {
        break;
    }
}
else
{
    printf("Y또는 N만 입력해주세요.₩n");
}
if(level==20)
{
```

강화게임

게임 화면


######<강화 시뮬레이션 프로그램입니다>######
@현재 컴퓨터 시간에 따라 랜덤숫자가 뽑히는데
@랜덤수가 강화 성공확률보다 낮으면 강화성공!
200 G | Level : 1
비용:50 G 95% 확률 | 강화하시겠습니까? Y/N

배열게임

Rule

1. 1~5 숫자를 중복 없이 랜덤으로 정렬한다.
2. 랜덤 생성된 숫자 배열을 1 2 3 4 5 로 10회 내에 완성 해야 한다.

배열게임

```
void arrnum(int* ptr)
{
    int i;
    for (i = 0; i < 5; i++)
    {
        printf("%d ", *(ptr + i));
    }
}
```

arrnum 함수 : 메인함수에서 선언된 arr배열의 값을 차례대로 출력해준다.
변형을 통해 바뀐 배열 값을 출력할 때 사용.

배열게임

```c
int main(void)
{
    int i,j,k, re = 1;
    int save, input;
    int arr[5];
    while (re == 1)
    {
        srand((int)time(NULL));

        for (i = 0; i < 5;) { //5ㅁ
            arr[i] = rand() % 5 + 1;
            for (j = 0, k = 0; j < i; j++) {
                if (arr[i] == arr[j])
                {
                    k = 1; break;
                }
            }
            if (k == 0)
                i = i + 1;

        }
```

i, j, k : 중복 없이 랜덤 생성을 위한 인덱스 변수
save, input : 배열 순서를 바꿀 때 사용 할 변수
re : 리플레이를 위한 변수
rand()%5 + 1 : 1~5 랜덤 변수 생성

배열게임

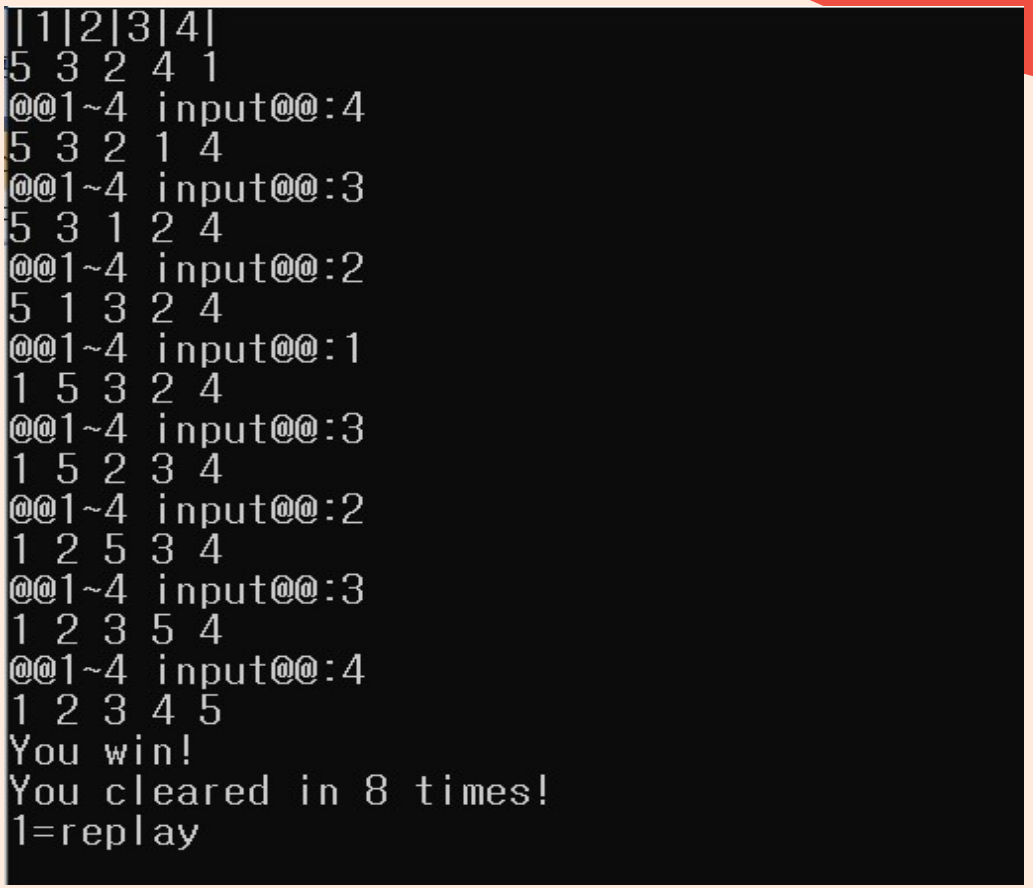
```
        re = 0;
        printf("|1|2|3|4|\n");
        arrnum(arr);
        for (i = 10; i > 0; i--)
        {
            save = 0;
            do
            {
                printf("\n@@1~4 input@@:");
                scanf("%d", &input);
            } while (input > 5 || input < 0);

            save = arr[input - 1];
            arr[input - 1] = arr[input];
            arr[input] = save;

            arrnum(arr);
            if ((arr[0] == 1 && arr[1] == 2) && ((arr[2] == 3 && arr[3] == 4) && arr[4] == 5))
            {
                printf("\nYou win!");
                printf("\nYou cleared in %d times!\n", 10 - i);
                break;
            }
        }
        if (i == 0)
        {
            printf("\nYou lose\n");
        }
        printf("1=replay");
        scanf("%d", &re);
    }
    return 0;
}
```

Input 변수에 1~4 중 하나를 입력하면 양 옆의 숫자 순서를 바꾼다.
순서대로 1 2 3 4 5가 완성되면 You win! 출력 후 몇 회 안에 성공했는지 출력한다.
만약 10회 안에 완성을 못 할 경우 You lose 출력.
게임 마무리 후 re 변수에 1 값을 입력하면 게임을 재시작 할 수 있다.

배열게임


|1|2|3|4|
5 3 2 4 1
@@1~4 input@@:4
5 3 2 1 4
@@1~4 input@@:3
5 3 1 2 4
@@1~4 input@@:2
5 1 3 2 4
@@1~4 input@@:1
1 5 3 2 4
@@1~4 input@@:3
1 5 2 3 4
@@1~4 input@@:2
1 2 5 3 4
@@1~4 input@@:3
1 2 3 5 4
@@1~4 input@@:4
1 2 3 4 5
You win!
You cleared in 8 times!
1=replay

야구게임

Rule

1. 3자리 숫자를 만든다 (1~9만 사용)

2. 위치 + 숫자가 맞다면 스트라이크,
숫자는 같지만, 위치가 다르면 볼,
숫자가 다르다면 아웃

야구게임

```c
int main()
{
    int answer[3];  //답을 저장할 배열 생성
    int player[3]; //사용자가 정답을 입력할 배열 생성
    int strike = 0, ball = 0, out = 0;  // strike 와 ball은 0으로 초기화한다.
    int i, j;
    int cnt = 0; //사용자가 입력한 횟수를 입력한다.
```

야구게임

```
for (i = 0; i < 3; i++) {
    srand(time(NULL)); //현재 시간을 이용해서 씨드 결정
    answer[i] = rand() % 9 + 1; // 1~9 사이의 숫자를 랜덤으로 answer[] 입력
    for (j = 0; j < i; j++) {
        if (answer[i] == answer[j]) { //answer[]배열의 중복된 숫자를 검사
            i--;
            break;
        }
    }
}
```

For 문 (I 가 0에서 2까지 총 3번 반복)
→ 3자리 수 이므로, 그 이상의 자릿수를 하려면 숫자 변경하면 됨

Srand(time(NULL)
→ 랜덤 값을 받아오는데, 시간에 대해서 받아오는 것
→ 이것을 사용하지 않으니, 프로그램을 여러 번 돌리니, 같은 값을 가짐..ㅜ

야구게임

```
for (i = 0; i < 3; i++) {
    srand(time(NULL)); //현재 시간을 이용해서 씨드 결정
    answer[i] = rand() % 9 + 1; // 1~9 사이의 숫자를 랜덤으로 answer[] 입력
    for (j = 0; j < i; j++) {
        if (answer[i] == answer[j]) { //answer[]배열의 중복된 숫자를 검사
            i--;
            break;
        }
    }
}
```

If문을 보면, i번째 자리와 j번째 자리가 서로 같은지 하나씩 확인 !

→ 중복된 숫자 사용 불가 예를 들면 777 같은 ..

야구게임

```
printf("₩n숫자 3개를 입력하세요(1 ~ 9까지의 숫자를 입력 숫자는 1칸씩 띄고 합니다.) : ");
scanf_s("%d %d %d", &player[0], &player[1], &player[2]);

if (player[0] == player[1] || (player[1] == player[2]) || (player[0] == player[2])) { //중복된 숫자를 검사
    printf("숫자는 중복을 허용하지 않습니다. 다시₩n");
    cnt--; //중복된 숫자를 입력했기 때문에 cnt(사용자 입력 횟수)를 다시 원래의 값으로 되돌린다.
    continue;
}
if (player[0] == 0 || player[1] == 0 || player[2] == 0) { // 0을 입력받지 못하도록 검사
    printf("숫자는 0을 허용하지 않습니다. ₩n");
    cnt--; //중복된 숫자를 입력했기 때문에 cnt(사용자 입력 횟수)를 다시 원래의 값으로 되돌린다.
    continue;
}
```

1. Scanf 를 통해 값을 받아옴
2. 이 값은 순서대로 player배열의 0번,1번,2번 자리를 차지
3. 앞선 과정과 같이 중복이 있는지 check !@!

야구게임

```
for (i = 0; i < 3; i++) {
    for (j = 0; j < 3; j++) {
        if ((answer[i] == player[j]) && i == j) { //컴퓨터가 입력한 숫자와 사용자가 입력한 숫자와 자리가 같을 경우 strike 값을 증가
            strike++;
        }
        if ((answer[i] == player[j]) && i != j) {  //같은 값이지만 자리가 다르면 ball값을 증가
            ball++;
        }
        if ((answer[i]) != player[j]) { //모든 숫자가 동일하지 않을 경우 out의 값을 증가
            out++;
        }
    }
}
```

1. i=j 일때, 즉 각 자리의 숫자가 같은경우, 스트라이크
2. 숫자는 같지만, 자리수가 다른경우, 볼
3. 숫자가 다르면, 아웃

THANK YOU