



게임 자료구조와 알고리즘

-CHAPTER2-

SOULSEEK



목차

1. 배열

2. 게임제작에 활용가능한 도구



1. 배열

1. 배열

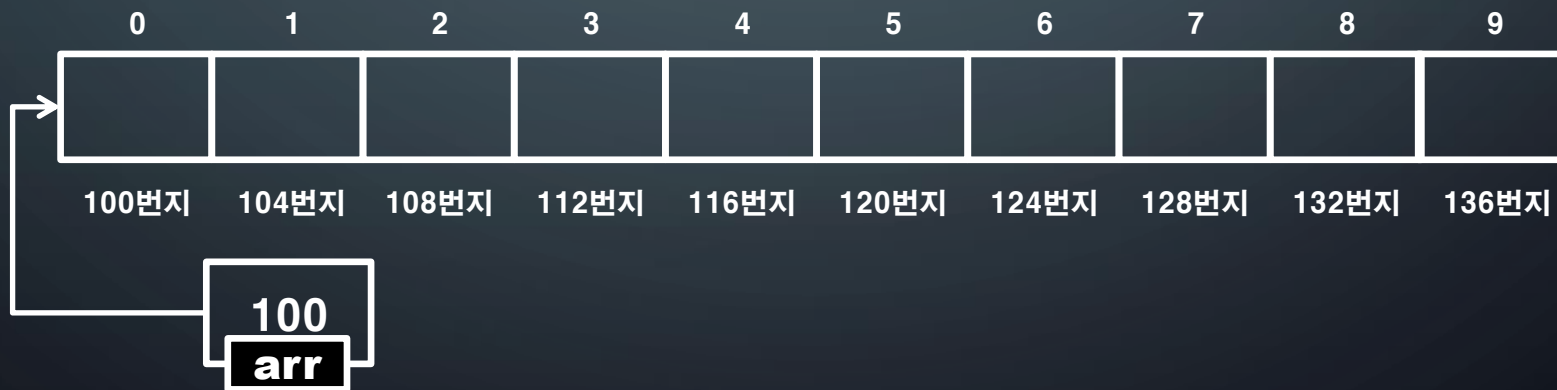
- 정의
 - **동일한 자료형으로 연속된 메모리 공간에 할당**하여 하나의 변수로 정보를 관리하는 것
 - 자료구조에서 가장 기본이 되는 요소, **선형자료구조**
- 형식
 - 자료형 배열명[배열길이]
 - 배열의 원소는 **0부터 시작**한다.
 - 배열명은 배열의 **시작주소**를 가진다.

Ex) **int Array[10];**
 char str[20];

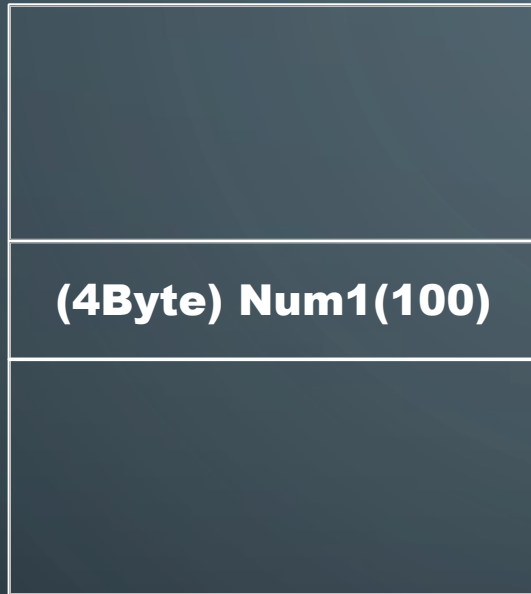
1. 배열

※ 배열 사용 예

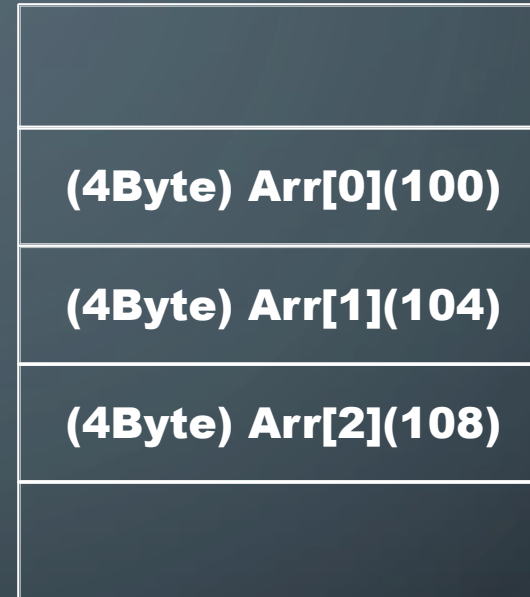
- `int arr[10]` 변수 선언
- `int arr[10] = {1,2,3,4,5,6,7,8,9,10}` 변수 선언 후 값 입력
- `int arr[10] = {0}` 변수 선언 후 초기화
- `Int arr[] = {1,2,3,4,5}` 변수 값 입력 후 크기 자동 지정
- 미리 지정한 크기보다 자동으로 설정하게 두면 후에 늘어남에 따라 사용이 편리 해진다.



1. 배열



int Num1



int Arr[3]

1. 배열

• 1차원 배열

```
#include <iostream>
using namespace std;
void main()
```

```
{
```

```
    int Arr[5];
```

```
    Arr[0] = 1;
```

```
    Arr[1] = 2;
```

```
    Arr[2] = 3;
```

```
    Arr[3] = 4;
```

```
    Arr[4] = 5;
```

```
    cout << Arr[0] << ", " << Arr[1] << ", " << Arr[2] << ", " << Arr[3] << ", " << Arr[4] << ",
    " << endl;
```

```
}
```



1. 배열

- 1차원 배열

```
#include <iostream>
using namespace std;
void main()
{
    int Arr[5];
    for (int i = 0; i < 5; i++)
    {
        Arr[i] = i + 1;
    }

    for (int i = 0; i < 5; i++)
    {
        cout << Arr[i] << endl;
    }
}
```



1. 배열

- 1차원 배열

```
#include <iostream>
using namespace std;
void main()
{
    int Arr[5];
    for (int i = 0; i < 5; i++)
    {
        cout << i + 1 << "번째 원소 값 : ";
        cin >> Arr[i];
    }

    for (int i = 0; i < 5; i++)
    {
        cout << Arr[i] << endl;
    }
}
```



1. 배열

- 1차원 배열(문자열)

```
#include <iostream>
using namespace std;
```

```
void main()
{
    char thisIsNotString[] = {'H', 'e', 'l', 'l', 'o' };

    char thisIsString[] = { 'H', 'e', 'l', 'l', 'o', '\0' };

    cout << thisIsNotString << endl;
    cout << thisIsString << endl;
}
```



1. 배열

• 1차원 배열(문자열)

```
#include <iostream>
using namespace std;
```

```
void main()
{
    char str[5];
    for (int i = 0; i < 5; i++)
    {
        cout << i + 1 << "번째 원소 값 : ";
        cin >> str;
    }

    for (int i = 0; i < 5; i++)
        cout << str[i];
}
```



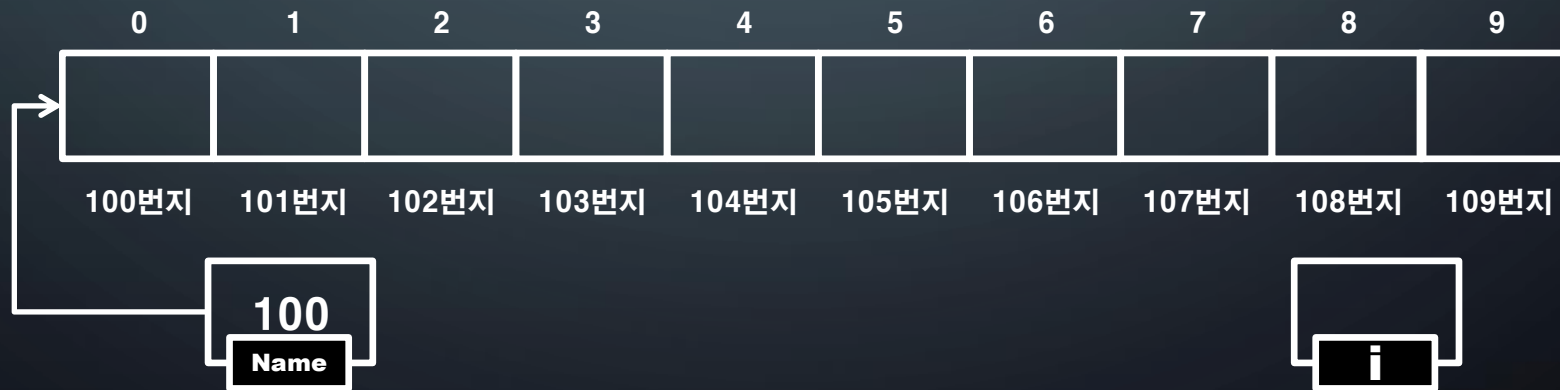
1. 배열

- 1차원 배열(문자열)

```
#include <iostream>
using namespace std;
#define SIZE 10
void main()
{
    char Name[SIZE];
    cin >> Name;

    for (int i = 0; i < SIZE; i++)
    {
        cout << Name[i];
    }

    cout << "\n" << "\n" << Name << endl;
}
```



1. 배열

연습문제

1. 아래의 코드들의 잘못된 점을 각각 설명해 보아라.

```
int arr[3];  
arr[1] = 10;  
arr[2] = 20;  
arr[3] = 30;
```

```
char arr[10];  
arr[20] = 'a';
```

```
float arr[5];  
arr = { 0.1f, 0.2f, 0.3f };
```

2. 다음 코드의 결과를 작성하라.

```
Int arr1[10];  
cout << "arr1[5] = " << arr1[15] << "\n";  
Int arr2[10] = { 1, 2, 3, 4, 5 };  
cout << "arr2[5] = " << arr2[5] << "\n";
```

3. 아래 배열의 크기를 byte 단위로 적어보자.

```
char c[5];  
char c[] = "Hello;"  
double d[3] ;  
short s[4];
```

학습과제_1

1. 배열의 길이가 10 int형 배열을 만들고 10개의 정수를 입력 받고 입력 받은 10개의 정수 중에서 가장 큰 수와 10개의 총합계, 평균을 출력하시요.
2. 다음 문자열을 뒤집어 출력하는 Code를 작성하시요.
 - "You are not ready yet"

1. 배열

배열을 활용한 게임처리

- 정렬 알고리즘을 활용한 랭킹 시스템
- 2차배열을 활용한 2차원 맵 처리 시스템

오름차순과 내림차순

- 특정 조건이 만족하는지를 데이터별로 비교해서 순위를 매겨준다.
- 0번째 원소부터 차례 대로 비교 대상자와 비교하여 조건에 맞는 쪽을 우선순위로 위치를 바꿔 준다.
- 오름 차순
 - 값의 크기가 점점 증가하며 정렬한다.
 - 1, 2, 3, 4, 5 or a, b, c, d
- 내림차순
 - 값의 크기가 점점 감소하며 정렬한다.
 - 5, 4, 3, 2, 1 or d, c, b, a

1. 배열

오름차순

```
#include<iostream>
using namespace std;
#define SIZE 5
```

```
void Ascending(int arr[])
{
    int tmp;
    for (int i = 0; i < SIZE - 1; i++)
    {
        for (int j = i + 1; j < SIZE; j++)
        {
            if (arr[i] > arr[j])
            {
                tmp = arr[i];
                arr[i] = arr[j];
                arr[j] = tmp;
            }
        }
    }
}
```

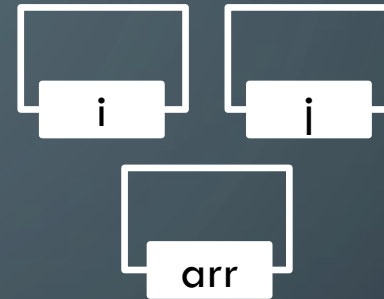
```
void main()
{
    int arr[SIZE] = { 4, 8, 2, 7, 6 };

    for (int i = 0; i < SIZE; i++)
        cout << arr[i];

    Ascending(arr);

    for (int i = 0; i < SIZE; i++)
        cout << arr[i];

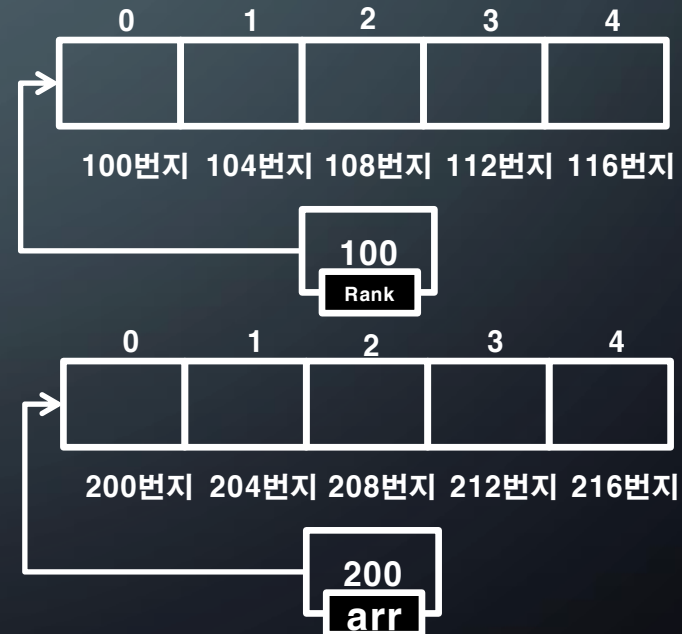
    cout << endl;
}
```



1. 배열 순위구하기

```
#define SIZE 5
void Ascending(int arr[],int Rank[])
{
    int Count;
    for (int i = 0; i < SIZE; i++)
    {
        Count = 1;
        for (int j = 0; j < SIZE; j++)
        {
            if (arr[i] < arr[j])
                Count++;
        }
        Rank[i] = Count;
    }
}

void main()
{
    int arr[SIZE] = { 82,85,76,79,96 };
    int Rank[SIZE];
    cout << "======" << endl;
    for (int i = 0; i < SIZE; i++)
        cout << arr[i] << "점 : ??등" << endl;
    cout << "======" << endl;
    Ascending(arr,Rank);
    cout << "======" << endl;
    for (int i = 0; i < SIZE; i++)
        cout << arr[i] << "점 : " << Rank[i] << "등" << endl;
    cout << "======" << endl;
}
```



학습과제_2

1. `int Arr[10]` 배열에 변수를 선언하여 그 안에 숫자를 입력 받고 최대로 입력 받은 뒤 내림차순으로 정렬하시요.
2. 입력 받은 숫자를 오름차순으로 정렬할 것인지 내림차순으로 정렬할 것인지 입력 받아서 결과를 출력하는 프로그램을 만드세요.
3. `int Arr[9]` 배열에 변수를 선언하여 그 안에 숫자를 입력 받고 중간순위의 값을 출력하고 그 숫자를 기준으로 큰 값과 작은 값을 차례로 나열하시요.

1. 배열

2차원 배열

- 1차원 배열을 여러 개 묶어서 만든 것.
- 메모리 구조상 똑같이 1열로 배열되겠지만 입체적인 표현을 하기위해 필요한 데이터 구조
- Arr[10] 과 Arr[5][2] 혹은 Arr[2][5] – Excel의 셀구조를 생각해보자.
- X, Y좌표의 2차원 개념이 들어가는 Map타일을 만들기 위한 구조.
 - 해당하는 배열의 숫자를 위치로 들어가는 값을 데이터를 판별하는 값으로 활용.

int Arr[10] = {0}

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

int Arr[5][5] = {0}

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

1. 배열

2차원 배열

```
void main()
```

```
{
```

```
    int arr[5][5] = { 0 };
```

```
    arr[2][2] = 1;
```

```
    for (int y = 0; y < 5; y++)
```

```
    {
```

```
        for (int x = 0; x < 5; x++)
```

```
        {
```

```
            cout << arr[y][x] << " ";
```

```
        }
```

```
        cout << endl;
```

```
    }
```

```
}
```

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	1	0	0
3	0	0	0	0	0
4	0	0	0	0	0

1. 배열

2차원 배열

```
#define STAR 1  
#define NULL 0
```

```
void main()  
{
```

```
    int arr[5][5] =
```

```
    {
```

```
        { 1, 1, 1, 1, 1 },
```

```
        { 1, 0, 0, 0, 1 },
```

```
        { 1, 0, 1, 0, 1 },
```

```
        { 1, 0, 0, 0, 1 },
```

```
        { 1, 1, 1, 1, 1 },
```

```
    };
```

```
    for (int y = 0; y < 5; y++)
```

```
    {
```

```
        for (int x = 0; x < 5; x++)
```

```
        {
```

```
            if (arr[y][x] == STAR)
```

```
            {
```

```
                cout << "★";
```

```
            }
```

```
            else if (arr[y][x] == NULL)
```

```
            {
```

```
                cout << "☆";
```

```
            }
```

```
        }
```

```
        cout << endl;
```

```
    }
```

```
}
```

	0	1	2	3	4
0	★	★	★	★	★
1	★	☆	☆	☆	★
2	★	☆	★	☆	★
3	★	☆	☆	☆	★
4	★	★	★	★	★

학습과제_3

1. 2차 배열을 만들고 0으로 초기화 한 후 원하는 X, Y 값을 받아서 입력 받은 곳을 특수 문자로 바꾸는 프로그램을 만드세요.
2. 1 ~ 25의 숫자를 본인이 원하는 위치 또는 랜덤으로 넣은 후 지속적으로 입력 받은 숫자와 동일한 곳의 값을 특수문자로 바꿔주는 프로그램을 만드세요.

2. 게임제작에 활용가능한 도구

2. 게임제작에 활용가능한 도구

Text 색깔 변경

- 지정한 순간부터 Text출력 색상이 변경하고 바꾸기 전까지 해당색깔이 적용된다.

```
#include<iostream>
#include<Windows.h>
using namespace std;

////////////////////////////////////
#define col GetStdHandle(STD_OUTPUT_HANDLE)
#define BLACK SetConsoleTextAttribute( col,0x0000 ); // 검정
#define DARK_BLUE SetConsoleTextAttribute( col,0x0001 ); // 검파랑
#define GREEN SetConsoleTextAttribute( col,0x0002 ); // 초록
#define BLUE_GREEN SetConsoleTextAttribute( col,0x0003 ); // 청록
#define BLOOD SetConsoleTextAttribute( col,0x0004 ); // 검붉은
#define PUPPLE SetConsoleTextAttribute( col,0x0005 ); // 보라
#define GOLD SetConsoleTextAttribute( col,0x0006 ); // 금색
#define ORIGINAL SetConsoleTextAttribute( col,0x0007 ); // 밝은 회색(원래 글자색)
#define GRAY SetConsoleTextAttribute( col,0x0008 ); // 회색
#define BLUE SetConsoleTextAttribute( col,0x0009 ); // 파랑
#define HIGH_GREEN SetConsoleTextAttribute( col,0x000a ); // 연두
#define SKY_BLUE SetConsoleTextAttribute( col,0x000b ); // 하늘
#define RED SetConsoleTextAttribute( col,0x000c ); // 빨강
#define PLUM SetConsoleTextAttribute( col,0x000d ); // 자주
#define YELLOW SetConsoleTextAttribute( col,0x000e ); // 노랑
#define WHITE SetConsoleTextAttribute(col, 0x000f); // 하양
////////////////////////////////////

void main()
{
    RED
        cout << "Hellow" << endl;
    ORIGINAL
}
```


위치 이동하는 특수문자 만들기 CODE 활용1

2. 게임제작에 활용가능한 도구

Sprintf 함수

- 문자열 사이에 서식문자를 이용해서 새로운 문자열로 재조합 할 때 사용한다.(stdio.h 필요)

```
#include<iostream>
#include<stdio.h>
using namespace std;
```

```
void main()
{
    char buf[40];
    int age = 20;
    char Name[10] = "SoulSeek";
    sprintf(buf, "%s님의 나이는 %d살 입니다.", Name, age);
    cout << buf << endl;
}
```

위치 이동하는 특수문자 만들기 CODE 활용2

게임만들기

- 2차원 배열의 학습내용을 활용해서 미로 찾기를 만들어 보자.

2. 게임제작에 활용가능한 도구

Sleep 함수

- 응용프로그램의 진행중 Millisecond 단위로 잠시 멈춘 뒤 다시 진행한다.(Windows.h)
- 반복문과 함께 일정 시간 뒤 정보가 갱신되는 update 역할을 할 수 있다.
 - 일정한 시간 뒤 정해진 위치로 이동

```
#include<iostream>
#include<Windows.h>
using namespace std;

void main()
{
    while(true)
    {
        system("cls");
        cout << "전화 거는중 ☎" << endl;

        for (int i = 0; i < 3; i++)
        {
            Sleep(1000);
            cout << ".";
        }

        Sleep(1000);
    }
}
```

일정시간이 흐른 뒤 이동하는 효과 CODE 활용

2. 게임제작에 활용가능한 도구

kbhit함수

- 키보드를 입력 했을 경우 참을 반환하는 함수(conio.h)

키 입력 상황을 체크해서 동작하는 CODE 활용

2. 게임제작에 활용가능한 도구

rand함수

- 임의의 수를 뽑아주는 함수(stdlib.h)
- `rand() % n`; n까지의 범위를 가지는 난수를 생성한다.

```
#include<iostream>
```

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

```
using namespace std;
```

```
void main()
```

```
{
```

```
    int RNum;
```

```
    while(1)
```

```
    {
```

```
        system("cls");
```

```
        for (int i = 1; i <= 6; i++)
```

```
        {
```

```
            RNum = rand();
```

```
            cout << i << "." << RNum << endl;
```

```
        }
```

```
        system("pause");
```

```
    }
```

```
}
```


2. 게임제작에 활용가능한 도구

srand 함수

- rand 함수의 갱신 조건을 설정해준다.

#include<time.h>//시간 값 사용하는 헤더파일

void main()

{

int RNum;

srand((unsigned)time(NULL));

while (1)

{

system("cls");

for (int i = 1; i <= 6; i++)

{

RNum = rand();

cout << i << "." << RNum << endl;

}

system("pause");

}

}

2. 게임제작에 활용가능한 도구

Rand함수 - 범위 설정

```
void main()
{
    int RNum;
    srand((unsigned)time(NULL));

    while (1)
    {
        system("cls");

        for (int i = 1; i <= 6; i++)
        {
            RNum = rand() % 45; // 0 ~ 44까지의 난수를 준다.
            //RNum = (rand() % 45) + 1; // 1 ~ 45까지의 난수를 준다.
            cout << i << "." << RNum << endl;
        }

        system("pause");
    }
}
```

The background is a dark blue gradient. In the corners, there are decorative white line art elements resembling circuit boards or neural network connections, with small circles at the end of the lines.

LOTTO RANDOM번호 받기 CODE 활용

The background is a dark blue gradient. In the corners, there are white line art illustrations of circuit boards or neural networks, with lines connecting to small circles.

눈탱이 맞는 확률 CODE 활용

게임만들기

1. Suffle 알고리즘 짜기 – CODE배끼면 뒤짐
2. 중복 수가 없는 Lotto 프로그램 만들기
3. UpDown게임 만들기(야구 게임처럼 세트 룰 적용)
4. 별똥별 피하기 게임 만들기