

민 선 생 코 딩 학 원   훈 련 반

---

# 수업노트 LV-17



# 배우는 내용

1. 비트배열로 마스킹하기
2. 다중 flag 및 다중 count
3. Base Index와 Offset 개념
4. getSum( ) 함수 만들기
5. isSame( ) 함수 만들기

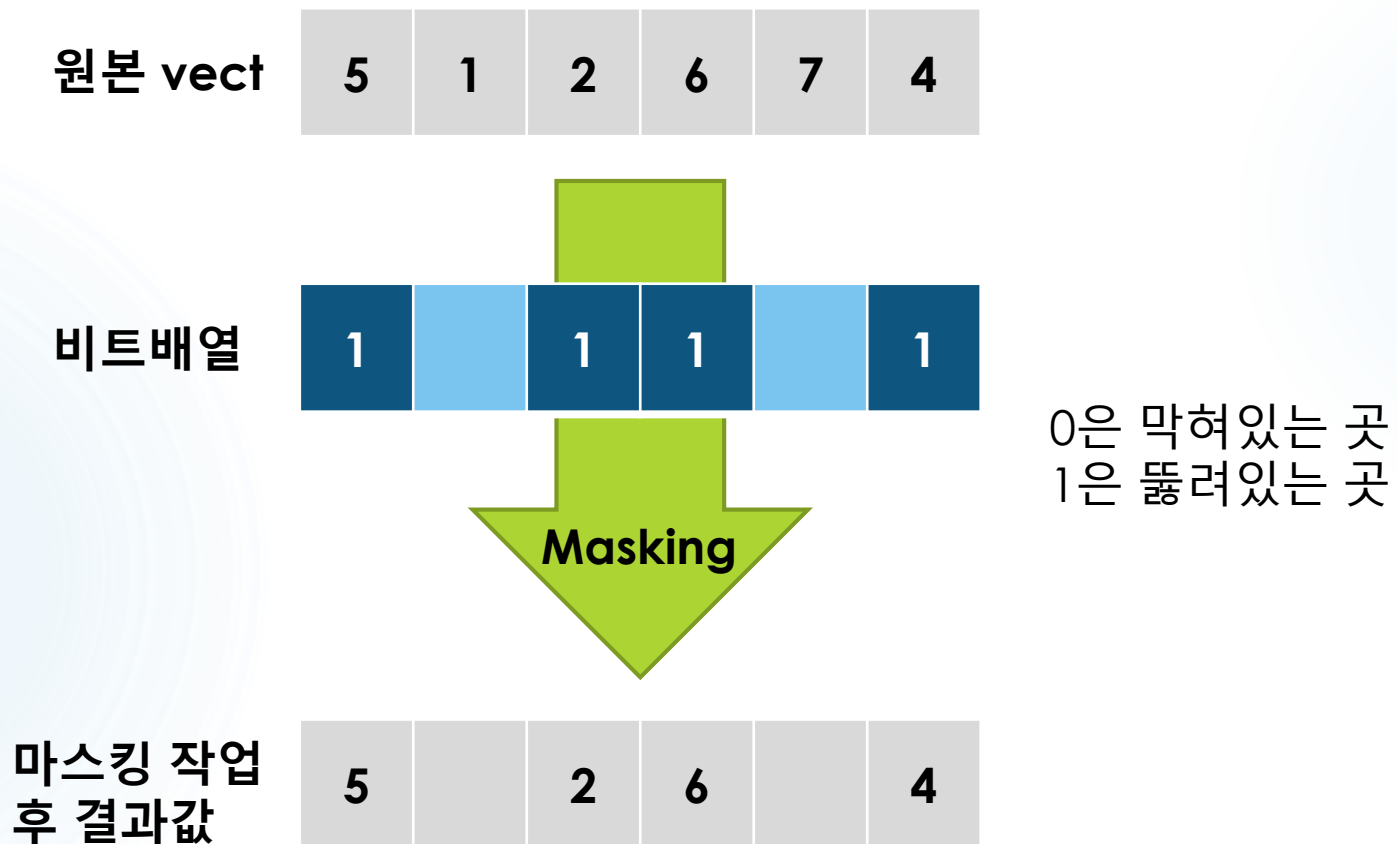
# 비트배열이란?

- ▶ 1과 0이 들어있는 배열
- ▶ 단순히 색칠해서 나타낼 수 있음



# 마스킹하기

- ▶ 내가 원하는 값만 뽑아 내기 위해, 필요 없는 값을 필터에 거르는 작업을 뜻함



# 마스킹하는 코드 예시

```
int main()
{
    int data[6] = { 5, 1, 2, 6, 7, 4 };
    int mask[6] = { 1, 0, 1, 1, 0, 1 };
    int result[6] = { 0 };

    int x;
    for (x = 0; x < 6; x++)
    {
        if (mask[x] == 1)
        {
            result[x] = data[x];
        }
    }

    return 0;
}
```

- ▶ result 배열에는  
마스킹 된 결과값이 저장 됨

# 다중 count

- ▶ 입력받은 숫자가 arr배열에 몇 개 있는지 Count하는 문제

arr     

1	5	1	5	3
---	---	---	---	---

input   

1	2	3
---	---	---

실행결과

1 : 2개

2 : 0개

3 : 1개

기존 count 기법을  
for로 묶으면 됨

```
for (y=0; y<3; y++)
{
    count = 0;
    for (x=0; x<5; x++)
    {
        if (arr[x] == input[y])
        {
            count++;
        }
    }
    cout << input[y] << " : " << count << "개\n";
}
```

# 다중 Flag

- 문자 3개를 입력받고, 입력받은 문자들이 arr배열에 각각 존재하는지 확인하는 프로그램

arr     A   D   C   A   R

input   A   Y   R

## 실행결과

A : 존재  
Y : 없음  
R : 존재

기존 flag 기법을  
for로 묶으면 됨

```
for (y=0; y<3; y++)  
{  
    flag = 0;  
    for (x=0; x<5; x++)  
    {  
        if (arr[x] == input[y])  
        {  
            flag = 1;  
            break;  
        }  
    }  
  
    if (flag == 1)  
    {  
        cout << input[y] << " : 존재" << endl;  
    }  
    else  
    {  
        cout << input[y] << " : 없음" << endl;  
    }  
}
```

[주의]  
y가 새롭게 시작 될 때마다  
flag를 초기화 해 주어야 함

# 다중 Flag 대신 isExist 함수 사용하기

- ▶ 다중 Flag :

2중 for문으로 구현해야 하기 때문에  
설계 및 코딩이 쉽지 않음

- ▶ isExist 함수 이용 :

main함수에서는 1중 for문으로 생각하면 되기 때문에  
코딩하기 간편 함

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

char arr[6] = "ADCAR";
char input[3];

int isExist(char ch)
{
    for (int x = 0; x < 6; x++)
    {
        if (arr[x] == ch)
        {
            return 1;
        }
    }
    return 0;
}

int main()
{
    for (int x = 0; x < 3; x++)
    {
        cin >> input[x];
    }

    for (int x = 0; x < 3; x++)
    {
        int ret = isExist(input[x]);

        if (ret == 1)
        {
            cout << input[x] << ":존재" << endl;
        }
        else
        {
            cout << input[x] << ":없음" << endl;
        }
    }

    return 0;
}
```



# Base Index와 Offset 개념

- ▶ Base Index는 기준 index라는 뜻이고  
Offset은 **기준 index로부터 얼마나 떨어져졌는지**를 나타내는 용어

a	b	c	d	e	f	g	h	i
---	---	---	---	---	---	---	---	---

base index를 1로 설정했을 때, 'd'의 offset 값은?  
→ [1]로부터 두칸 떨어져 있으므로 offset은 **2** 이다.

base index가 'f'의 index일 때 , 'a'의 offset 값은?  
→ offset은 **-5** 이다

## 2차원 배열에서 offset

- ▶ base 좌표를 입력받고  
y축으로 +1, x축으로 -2 떨어진 값 출력

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

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

int vect[6][4] = {
    3, 4, 4, 7,
    1, 5, 6, 7,
    3, 6, 6, 9,
    2, 7, 9, 0,
    5, 3, 3, 1,
    1, 6, 8, 4
};

int main()
{
    int dy, dx;

    cin >> dy >> dx;

    cout << vect[dy + 1][dx - 2];

    return 0;
}
```

### [참고] {} 생략 가능

단, 배열사이즈를  
vect[6][5] 이렇게 만들면  
값이 잘못 들어가게 됨을 주의

# getSum 함수 만들기1

- ▶ index 1개를 입력받고, 연속된 4개의 배열 값들의 합 출력하는 프로그램

4	3	5	1	7	5	6	8	2
---	---	---	---	---	---	---	---	---

- ▶ 예로들어 1을 입력받으면  $3 + 5 + 1 + 7 = 16$  출력
- ▶ `int getSum(int target)` 함수를 이용해서 문제를 해결

# getSum 함수 만들기2

- ▶ index 1개를 입력받고,  
연속된 4개의 배열 값들의 합 출력하는 프로그램

4	3	5	1	7	5	6	8	2
---	---	---	---	---	---	---	---	---

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

int vect[9] = {4, 3, 5, 1, 7, 5, 6, 8, 2};

int getSum(int index)
{
    int sum = 0;
    for (int x = index; x < index + 4; x++)
    {
        sum += vect[x];
    }
    return sum;
}

int main()
{
    int index;

    cin >> index;

    int ret = getSum(index);
    cout << ret;

    return 0;
}
```

# 두 배열이 완전히 똑같은지 확인하는 방법 <중요> - Flag

1중 for를 돌려서 완전히 똑같은 배열인지 확인하면 된다.  
**flag**를 쓰는 방법이 있고, **is**함수를 쓰는 방법이 있다  
먼저 **Flag**를 쓰는방법

vect1

f	g	h	i
---	---	---	---

vect2

f	g	h	i
---	---	---	---

다른글자가 하나라도 발견시  
flag를 1로 켜고 for를 종료

```
#include <stdio.h>

char data1[5] = "fghi";
char data2[5] = "fghi";

int main()
{
    int x;
    int flag = 0;
    for (x = 0; x < 4; x++)
    {
        if (data1[x] != data2[x])
        {
            flag = 1;
            break;
        }
    }

    if (flag == 0) printf("동일");
    else printf("동일하지 않음");

    return 0;
}
```

# 두 배열이 완전히 똑같은지 확인하는 방법 <중요> - Is 함수

“같습니까?”라는 의미의  
함수 이름이기 때문에  
완전 같은 경우는 return 1  
다른 경우는 return 0

```
char data1[5] = "fghi";  
char data2[5] = "fghi";  
  
int IsSame()  
{  
    int x;  
  
    for (x = 0; x < 4; x++)  
    {  
        if (data1[x] != data2[x])  
        {  
            return 0;  
        }  
    }  
  
    return 1;  
}  
  
int main()  
{  
    int result = IsSame();  
  
    if (result == 1) printf("동일");  
    else printf("동일하지 않음");  
  
    return 0;  
}
```

다른 글자가 하나라도 발견시  
바로 return 0;

여기까지 코드가 오면  
다른글자가 하나도 발견 되지 않았으니  
return 1;