

최범수 3일차 과제

1. HW_001

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

int main()
{
    //입력받을 최솟값과 최댓값 초기화, cnt는 제곱수 셀 변수 초기화
    int min = 0, max = 0;
    int cnt = 0;
    int k, i = 0, j;
    //cnt_result는 전체 수에서 제곱수 제외하고 셀 변수
    int cnt_result = 0;
    int arr[10000] = { 0, };

    printf("min : ");
    scanf("%d", &min);
    printf("max : ");
    scanf("%d", &max);

    //i는 최소부터 최대까지
    for ( i = min; i <= max; i++)
    {
        //j는 1을 제외하고 2부터 제곱해서 i를 나누었을 때 나머지가 0인지 아닌지
        //비교하기 위해 선언
        for ( j = 2; j <= i; j++)
        {
            if ( i % ( j * j ) == 0 )
            {
                //제곱수의 개수를 세는 과정 한번 나눠지면 break
                cnt++;
                break;
            }
        }
        //과정이 끝나고 만약 제곱수로 나눠지지 않으면 배열 arr에 저장
        if ( i % ( j * j ) != 0 )
        {
            arr[i] = i;
        }
    }
    //최종 결과는 최대에서 최소 빼고 1을 더한 뒤 제곱수 개수만큼 빼기
    cnt_result = (max - min + 1) - cnt;
    printf("제곱 수 : %d개", cnt_result);
    printf("\n");

    //총 개수를 먼저 출력하고 그 다음에 제곱으로 나눠지지 않은 수를 출력하는 과정 윗부분과
    조건은 유사
    for ( i = min; i <= max; i++)
    {
        if ( i % ( j * j ) != 0 && arr[i] != 0 )
    }
```

```

        {
            printf("%d ", arr[i]);
        }
    }
}

```

```

#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

int main()
{
    //입력받을 최소값과 최대값 초기화
    int min = 0, max = 0;
    int cnt = 0;
    int k, i = 0, j;
    //cnt_result는 전체 수에서 제공
    int cnt_result = 0;
    int arr[10000] = { 0, };

    printf("min : ");
    scanf("%d", &min);
    printf("max : ");
    scanf("%d", &max);

    //i는 최소부터 최대까지

```

Microsoft Visual Studio 디버그

```

min : 1
max : 10
제공된 수 : 7개
1 2 3 5 6 7 10
C:\Users\chlqj\source\repos\HW_07_03\x64
디버깅이 중지될 때 콘솔을 자동으로 닫으
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

```

#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

int main()
{
    //입력받을 최소값과 최대값 초기화
    int min = 0, max = 0;
    int cnt = 0;
    int k, i = 0, j;
    //cnt_result는 전체 수에서 제공수
    int cnt_result = 0;
    int arr[10000] = { 0, };

    printf("min : ");
    scanf("%d", &min);
    printf("max : ");
    scanf("%d", &max);

    //i는 최소부터 최대까지

```

Microsoft Visual Studio 디버그

```

min : 3
max : 12
제공된 수 : 6개
3 5 6 7 10 11
C:\Users\chlqj\source\repos\HW_07_03\x64
디버깅이 중지될 때 콘솔을 자동으로 닫으
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

Min이 1이 아닌 다른 숫자여도 실행되게 제작했고 출력 과정에서 개수와 제공으로 나누어지지 않는 수가 반대로 출력돼서 똑 같은 양식의 for문을 두번 돌려 순서를 조정했습니다.

2. HW_002

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <string.h>

//strcmp 함수 사용
int strcmp();
//전역변수 num, cnt 초기화
int num, cnt = 0;
//문자열 입력받을 배열 초기화
char input[] = {" "};
//집합을 출력해 줄 함수 선언
int printprint(int save_print[])
{
    //처음에 20보다 작게 했다가 20이 출력이 안돼서 20보다 작거나 같자로 수정
    for (int i = 0; i <= 20; i++)
    {
        //남아 있는 집합 숫자를 save_print에 저장했는데 저장되지 않은 부분을 0으로
        //설정해서 0이 아닌 숫자만 출력되게 설정
        if (save_print[i] != 0)
        {
            printf("%d, ", save_print[i]);
        }
    }
}

int main()
{
    //처음에 조건을 0부터 20미만으로 했었는데 20이 출력이 안돼서 배열을 21칸으로 만들고
    //20까지 출력 앞에 0은 출력안되게 설정
    int save[21] = { 0, };
    int count = 0;
    //예시 부분
    printf("연산을 선택하세요. ( 1 <= x <= 20 )\n");
    printf("add X\n");
    printf("remove X\n");
    printf("check X\n");
    printf("toggle X\n");
    printf("all 0\n");
    printf("empty 0\n");
    printf("X\n");
    //무한 반복되게 설정
    while (1)
    {
        printf("input : ");
        scanf("%s", &input); //input인지 remove인지 즉 연산 선택
        //입력받은 것이 add
        if (!strcmp(input, "add"))
        {
            //처음에 위 scanf와 같이 입력받았으나 뒤에 all 0와 empty 0의 0을
            //문자열과 같이 인식 못해서 각 연산 안에서 숫자 입력받음
            scanf("%d", &num);
        }
    }
}
```

아니라고 판단

```
printf("집합 : { ");
//숫자가 1부터 20까지 일 때만 실행
if (num >= 1 && num <= 20)
{
    //저장된 숫자가 방금 입력받은 것과 같으면 continue
    for (int i = 0; i < 20; i++)
    {
        if (save[i] == num)
        {
            continue;
        }
        //다르다면 cnt를 1씩 증가시켜서 총 20이 되면 아무것도 중복이

    else if (save[i] != num)
    {
        cnt++;
    }
    //중복이 안된경우
    if (cnt == 20)
    {
        save[count] = num;
        //count 숫자를 1 증가해서 배열 1칸 이동
        count++;
        //출력함수에 저장된 숫자 출력
        printprint(save);
        //다시 cnt를 0으로 초기화해서 add를 여러번 사용 가능
        cnt = 0;
    }
}

printf(" }");
printf("\n");
printf("\n");
}
//remove
else if (!strcmp(input, "remove"))
{
    scanf("%d", &num);
    printf("집합 : { ");
    //숫자를 1부터 20까지로 설정
    if (num >= 1 && num <= 20)
    {
        //i를 0부터 20미만까지로 설정
        for (int i = 0; i < 20; i++)
        {
            //저장된 숫자가 입력받은 숫자랑 같으면 저장된 숫자를
            if (save[i] == num)
            {
                save[i] = 0;
            }
        }
        //그 후 중복이 없으면 그대로 출력
    }
}
```

```

        printprint(save);
    }
    printf(" }");
    printf("\n");
    printf("\n");
}
//check
else if (!strcmp(input, "check"))
{
    scanf("%d", &num);
    // 숫자를 1부터 20까지로 설정
    if (num >= 1 && num <= 20)
    {
        //check용 카운트 변수 선언
        int cnt_check = 0;
        for (int k = 0; k < 20; k++)
        {
            //저장된 숫자가 입력받은 숫자와 같으면 1출력
            if (save[k] == num)
            {
                printf("1 ");
                printf("집합 : {");
                printprint(save);
                printf(" }");
                printf("\n");
                printf("\n");
            }
            //다르면 check를 1씩 증가시켜서 총 20이 cnt에

            else if (save[k] != num)
            {
                cnt_check++;
            }
        }
        //0으로 설정되면 출력 안되게 설정
        if (cnt_check == 20)
        {
            printf("0 ");
            printf("집합 : {");
            printprint(save);
            printf(" }");
            printf("\n");
            printf("\n");
        }
    }
}
//toggle
else if (!strcmp(input, "toggle"))
{
    scanf("%d", &num);

    printf("집합 : { ");
    if (num >= 1 && num <= 20)
    {

```

찍한다면 0으로 초기화

거니까 출력

안되게 설정

```
//toggle용 cnt 설정
int cnt_toggle = 0;
for (int k = 0; k < 20; k++)
{
    //저장받은 숫자와 입력받은 숫자가 같은지 판단
    if (save[k] == num)
    {
        continue;
    }
    //다르면 toggle을 1씩 증가해서
    else if (save[k] != num)
    {
        cnt_toggle++;
    }
    //총 20번 반복해서 cnt가 20이 된다면 중복이 없는
}
if (cnt_toggle == 20)
{
    save[count] = num;
    count++;
    printprint(save);
}
//만약 20이 아니라면 중복인 거니까 0으로 초기화해서 출력

else if (cnt_toggle != 20)
{
    for (int k = 0; k < 20; k++)
    {
        if (save[k] == num)
        {
            save[k] = 0;
        }
    }
    printprint(save);
}

}
printf(" }");
printf("\n");
printf("\n");
}
//all 0
else if (!strcmp(input, "all"))
{
    //이 연산 때문에 숫자를 연산 안에서 입력받게 설정
    scanf("%d", &num);

    printf("집합 : { ");
    // 0을 입력받았다면 1부터 20까지의 숫자를 배열에 저장시킨 뒤 출력
    if (num == 0)
    {
        for (int k = 0; k <= 20; k++)
        {
            save[k] = k;
        }
    }
}
```

```

        printprint(save);
    }
    printf(" }");
    printf("\n");
    printf("\n");
}
else if (!strcmp(input, "empty"))
{
    scanf("%d", &num);

    printf("집합 : { ");
    //0을 입력받았다면 모든 배열에 숫자를 0으로 초기화하고 출력
    if (num == 0)
    {
        for (int k = 0; k <= 20; k++)
        {
            save[k] = 0;
        }
        printprint(save);
    }
    printf(" }");
    printf("\n");
    printf("\n");
}
//이 연산이 아닌 다른 부분을 작성한다면 연산 및 숫자를 다시 입력
else
{
    printf("연산 및 숫자를 다시 입력해주세요.");
}
}
}

```

The screenshot shows a C++ IDE with a dark theme. The left pane displays the source code, and the right pane shows the program's output.

Source Code (Left Pane):

```

//전역 범위
else if (save[k] != num)
{
    cnt_check++;
}
//0으로 설정되면 출력 안되게 설정
if (cnt_check == 20)
{
    printf("0 ");
    printf("집합 : {");
    printprint(save);
    printf(" }");
    printf("\n");
    printf("\n");
}
}
}

```

Program Output (Right Pane):

```

연산을 선택하세요. ( 1 <= x <= 20 )
add X
remove X
check X
toggle X
all 0
empty 0

input : add 5
집합 : { 5, }

input : add 3
집합 : { 5, 3, }

input : remove 5
집합 : { 3, }

input : check 3
1 집합 : {3, }

input : toggle 3
집합 : { }

input : all 0
집합 : { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, }

input : empty 0
집합 : { }

input :

```

```
stdlio.h HW_07_03_02.c HW_07_03_01.c (전역 범위)
    save[k] = 0;
    printprint(save);
}
printf(" }");
printf("\n");
printf("\n");
}
//이 연산이 아닌 다른 부분을 작성한
else
{
    printf("연산 및 숫자를 다시 입력하세요.");
}
}

연산을 선택하세요. ( 1 <= x <= 20 )
add X
remove X
check X
toggle X
all 0
empty 0

input : add 5
집합 : { 5, }

input : toggle 6
집합 : { 5, 6, }

input : toggle 5
집합 : { 6, }

input : toggle 6
집합 : { }

input : empty 0
집합 : { }

input : all 0
집합 : { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, }

input : remove 4
집합 : { 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, }

input : |
```

all로 모두 출력하고 remove로 지우기 toggle로 입력 및 지우기

```
stdlio.h HW_07_03_02.c HW_07_03_01.c (전역 범위)
    for (int k = 0; k <= 20; k++)
    {
        save[k] = k;
        printprint(save);
    }
    printf(" }");
    printf("\n");
    printf("\n");
}
else if (!strcmp(input, "empty"))
{
    scanf("%d", &num);
    printf("집합 : { ");
    //0을 입력받았다면 모든 배열이 0이 되도록 초기화
    if (num == 0)
    {
        for (int k = 0; k <= 20; k++)
        {
            save[k] = 0;
        }
    }
    printprint(save);
}

연산을 선택하세요. ( 1 <= x <= 20 )
add X
remove X
check X
toggle X
all 0
empty 0

input : add 3
집합 : { 3, }

input : remove 3
집합 : { }

input : remove 3
집합 : { }

input : toggle 3
집합 : { 3, }

input : toggle 3
집합 : { }

input : all 0
집합 : { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, }

input : toggle 5
집합 : { 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, }

input : |
```

Remove 반복 시키기 toggle로 출력하고 지우기 all로 모두 출력했다 toggle로 지우기

The image shows a C++ IDE with two panels. The left panel displays the source code for a set implementation. The right panel shows the program's output, which is a sequence of commands and the current state of the set.

```
for (int k = 0; k <= 20; k++)
{
    save[k] = k;
}
printf(" ");
printf("\n");
printf("\n");
}

else if (!strcmp(input, "empty"))
{
    scanf("%d", &num);
    printf("집합 : { ");
    //0을 입력받았다면 모든 배열에 숫자를
    if (num == 0)
```

연산을 선택하세요. (1 <= x <= 20)

add X
remove X
check X
toggle X
all 0
empty 0

input : remove 3
집합 : { }

input : add 3
집합 : { 3, }

input : toggle 3
집합 : { }

input : empty 0
집합 : { }

input : add 5
집합 : { 5, }

input : |

아무것도 없는 상태에서 3 지우고 add로 넣었다고 toggle로 지우고 빈 상태에서 empty 쓰고 다시 add로 5 넣기