

# 서현규 3일차 과제

## 1. HW\_001

### 소스코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main() {
    int min, max, count=0, nNumcnt = 0;
    int num[10000];
    printf("min :");
    scanf("%d", &min);
    printf("max :");
    scanf("%d", &max);

    for (int i = min; i <= max; i++) { //min부터 max 사이의 숫자 개수만큼 반복
        //min부터 max 사이의 숫자(i)들을 자연수(2부터 1씩 증가함)의 제곱으로 나눠
        나머지가 0이 아니면 배열에다 집어넣고 카운트를 하는 코드
        for (int nNum = 2; nNum < max; nNum++) {
            if (i % (nNum * nNum) != 0) { //어떤 제곱수로 나눠 나머지가 모두
0이 아니었다면
                                nNumcnt++; //그 횟수를 센다
                                if (nNumcnt == max-2) { //모든 제곱수로 나눠 나머지가 0이
아니었다면
                                    num[count] = i; //제곱ㄴㄴ수이므로 배열에 넣고
count에 1만큼 더한다
                                    count++;
                                }
                            }
                        }
                    }
                nNumcnt = 0; //다음 숫자를 판정하기 위해 제곱수로 나눠 나머지가 0이 아닌
                횟수를 센 값을 초기화시켜준다
            }

            printf("제곱 ㄴㄴ수 : %d개\n", count); //카운트한 수와 배열을 출력한다
            for (int i = 0; i < count; i++) {
                printf("%d ", num[i]);
            }
        }
    }
```

## 실행결과

```
Microsoft Visual Studio 디버그 × + v
min :1
max :10
제공 ㄴㄴ 수 : 7개
1 2 3 5 6 7 10
C:\Users\e5gyg\source\repos\HW0703\x64\Deb
디버깅이 중지될 때 콘솔을 자동으로 닫으려면
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...|
```

```
Microsoft Visual Studio 디버그 × + v
min :3
max :20
제공 ㄴㄴ 수 : 11개
3 5 6 7 10 11 13 14 15 17 19
C:\Users\e5gyg\source\repos\HW0703\x64\Deb
디버깅이 중지될 때 콘솔을 자동으로 닫으려면
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...|
```

## 2. HW\_002

### 소스코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>

int ziphap[20] = {0, }; //배열의 모든 값에 0을 넣음

int add(int x);
int remove(int x);
int check(int x);
int toggle(int x);
int all(int x);
```

```
int empty(int x);
```

```
int main() {  
    int x;  
    int size;  
    char command[10];  
    printf("연산을 선택하세요. (1 <= x <= 200\n");  
    printf("add X\nremove X\n check X\ntoggle X\nall 0\nempty 0\n\n");  
    while (1) { //무한 루프  
        printf("input : ");  
        scanf("%s %d", &command, &x);
```

실행하는 코드

```
        if (!strcmp(command, "add")) { //command를 입력한 값을 판정해서 함수를
```

```
            x = add(x);  
        }  
        if (!strcmp(command, "remove")) {  
            x = remove(x);  
        }  
        if (!strcmp(command, "check")) {  
            x = check(x);  
        }  
        if (!strcmp(command, "toggle")) {  
            x = toggle(x);  
        }  
        if (!strcmp(command, "all")) {  
            x = all(x);  
        }  
        if (!strcmp(command, "empty")) {  
            x = empty(x);  
        }  
    }
```

```
    printf("집합 : {"); //출력  
    for (int n = 0; n < 20; n++) { //x의 범위에 0은 포함되지 않는 것을 이용해서  
0은 출력되지 않게 코드를 짤
```

```
        if (ziphap[n] != 0) {  
            printf("%d, ", ziphap[n]);  
        }  
    }
```

```
    printf("}\n\n");  
}
```

```
}
```

```
int add(int x){ //add함수
```

```
    for (int i = 0; i < 20; i++) {  
        if (ziphap[i] == 0 || ziphap[i] == x) { //집합에 빈칸이 있거나 x와 같은 값이  
있다면
```

```
            ziphap[i] = x; //거기에다 x 덮어씌우고 탈출  
            break;  
        }  
    }
```

```
    }  
}
```

```

int remove(int x){//remove함수
    for (int i = 0; i < 20; i++) {
        if (ziphap[i] == x) {//x와 같은 값이 있으면 없애기
            ziphap[i] = 0;
            for (int j = 0; j < i + 1; j++) {//그 뒤부터 배열의 모든 칸을 한 칸
                ziphap[i + j] = ziphap[i + j + 1];
                ziphap[i + j + 1] = 0;
            }
            break;
        }
    }
}

int check(int x){//check함수
    int i; //for문 밖에도 i를 써야해서 아예 따로 선언함
    for (i = 0; i < 20; i++) {
        if (ziphap[i] == x) {
            printf("1 "); //x와 같은 값이 있으면 1 출력하고 탈출
            break;
        }
    }
    if (i == 20) {//x와 같은 값이 없어서 for문이 최대를 반복했을 경우
        printf("0 ");
    }
}

int toggle(int x){//toggle함수
    for (int i = 0; i < 20; i++) {
        if (ziphap[i] == x) {
            ziphap[i] = 0;
            for (int j = 0; j < i + 1; j++) {
                ziphap[i + j] = ziphap[i + j + 1];
                ziphap[i + j + 1] = 0;
            }
            break;
        }
        else if (ziphap[i] == 0) {
            ziphap[i] = x;
            break;
        }
    }
}

int all(int x){//all함수
    for (int i = 0; i < 20; i++) {
        ziphap[i] = i+1;
    }
}

int empty(int x){//empty함수
    for (int i = 0; i < 20; i++) {
        ziphap[i] = 0;
    }
}

```

## 실행결과

```
C:\Users\We5gyg\source\Wrep  ×  +  ▾  
연산을 선택하세요. (1 ≤ x ≤ 200  
add X  
remove X  
  check X  
toggle X  
all 0  
empty 0  
  
input : add 1  
집합 : {1, }  
  
input : add 2  
집합 : {1, 2, }  
  
input : add 10  
집합 : {1, 2, 10, }  
  
input : remove 2  
집합 : {1, 10, }  
  
input : check 10  
1 집합 : {1, 10, }  
  
input : check 1  
1 집합 : {1, 10, }  
  
input : check 9  
0 집합 : {1, 10, }  
  
input : toggle 1  
집합 : {10, }  
  
input : toggle 1  
집합 : {10, 1, }  
  
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 : |
```