

과제1

어떤 수 X가 1보다 큰 제곱수로 나누어 떨어지지 않을 때, 제곱ㄴㄴ수라고 한다.

제곱수는 4, 9, 16, 25와 같은 것이고, 제곱ㄴㄴ수는 1,2,3,5,6,7,10,11,13,...과 같은 수이다.

min과 max가 주어지면, min과 max를 포함한 사이에 제곱ㄴㄴ수가 몇 개 있는지 출력한다.

입력은 정수 1 ~ 10000사이로 한다.

```
#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]);
    }
}
```

테스트 케이스

```
Microsoft Visual Studio 디버그 × + ▾

min :1
max :10
제곱 ㄴㄴ수 : 7개
1 2 3 5 6 7 10
C:\Users\e5gyg\source\repos\HW0703\x64\Deb
디버깅이 중지될 때 콘솔을 자동으로 닫으려만
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요 ...|
```

과제 2

비어있는 공집합 S가 주어졌을 때, 아래 연산을 수행하는 프로그램을 작성하시오.

- add x: S에 x를 추가한다. ($1 \leq x \leq 20$) S에 x가 이미 있는 경우에는 연산을 무시한다.
- remove x: S에서 x를 제거한다. ($1 \leq x \leq 20$) S에 x가 없는 경우에는 연산을 무시한다.
- check x: S에 x가 있으면 1을, 없으면 0을 출력한다.
- toggle x: S에 x가 있으면 x를 제거하고, 없으면 x를 추가한다. ($1 \leq x \leq 20$)
- all 0: S를 {1, 2, ..., 20} 으로 바꾼다.
- empty 0: S를 공집합으로 바꾼다.

```
HW0703 (전역 범위)
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <string.h>
4
5  int ziphap[20] = {0, }; //배열의 모든 값에 0을 넣음
6
7  int add(int x);
8  int remove(int x);
9  int check(int x);
10 int toggle(int x);
11 int all(int x);
12 int empty(int x);
13
14 int main() {
15     int x;
16     int size;
17     char command[10];
18     printf("연산을 선택하세요. (1 <= x <= 20)\n");
19     printf("add %d\nremove %d\ncheck %d\ntoggle %d\nall 0\nempty 0\n\n");
20     while (1) { //무한 루프
21         printf("input : ");
22         scanf("%s %d", &command, &x);
23
24         if (!strcmp(command, "add")) { //command를 입력한 값을 판정해서 함수를 실행하는 코드
25             x = add(x);
26         }
27         if (!strcmp(command, "remove")) {
28             x = remove(x);
29         }
30         if (!strcmp(command, "check")) {
31             x = check(x);
32         }
33         if (!strcmp(command, "toggle")) {
34             x = toggle(x);
35         }
36         if (!strcmp(command, "all")) {
37             x = all(x);
38         }
39     }
40 }
```

```

38     }
39     if (!strcmp(command, "empty")) {
40         x = empty(x);
41     }
42
43     printf("집합 : {"); //출력
44     for (int n = 0; n < 20; n++) { //x의 범위에 0은 포함되지 않는 것을 이용해서 0은 출력되지 않게 코드를 짰
45         if (ziphap[n] != 0) {
46             printf("%d, ", ziphap[n]);
47         }
48     }
49     printf("\n\n");
50 }
51 }
52
53 int add(int x) { //add함수
54     for (int i = 0; i < 20; i++) {
55         if (ziphap[i] == 0 || ziphap[i] == x) { //집합에 빈칸이 있거나 x와 같은 값이 있다면
56             ziphap[i] = x; //거기에다 x 넣어둬주고 탈출
57             break;
58         }
59     }
60 }
61 int remove(int x) { //remove함수
62     for (int i = 0; i < 20; i++) {
63         if (ziphap[i] == x) { //x와 같은 값이 있으면 없애기
64             ziphap[i] = 0;
65             for (int j = 0; j < i + 1; j++) { //그 뒤부터 배열의 모든 칸을 한 칸 당기기
66                 ziphap[i + j] = ziphap[i + j + 1];
67                 ziphap[i + j + 1] = 0;
68             }
69             break;
70         }
71     }
72 }

```

```

72 }
73 int check(int x){//check함수
74     int i;//for문 밖에도 i를 써야해서 아예 따로 선언함
75     for (i = 0; i < 20; i++) {
76         if (ziphap[i] == x) {
77             printf("1 ");//x와 같은 값이 있으면 1 출력하고 탈출
78             break;
79         }
80     }
81     if (i == 20) {//x와 같은 값이 없어서 for문이 최대를 반복했을 경우
82         printf("0 ");
83     }
84 }
85 int toggle(int x){//toggle함수
86     for (int i = 0; i < 20; i++) {
87         if (ziphap[i] == x) {
88             ziphap[i] = 0;
89             for (int j = 0; j < i + 1; j++) {
90                 ziphap[i + j] = ziphap[i + j + 1];
91                 ziphap[i + j + 1] = 0;
92             }
93             break;
94         }
95         else if (ziphap[i] == 0) {
96             ziphap[i] = x;
97             break;
98         }
99     }
100 }
101 int all(int x){//all함수
102     for (int i = 0; i < 20; i++) {
103         ziphap[i] = i+1;
104     }
105 }

```

```

106 int empty(int x){//empty함수
107     for (int i = 0; i < 20; i++) {
108         ziphap[i] = 0;
109     }
110 }

```

테스트 케이스

C:\Users\We5gyg\source\Wref

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

```
input : add 5
집합 : {5, }
```

```
input : add 3
집합 : {5, 3, }
```

```
input : check 3
1 집합 : {5, 3, }
```

```
input : toggle 3
집합 : {5, }
```

```
input : toggle 3
집합 : {5, 3, }
```

```
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 : |
```

#include <string.h>