

Euidong Lee
Junseok Lee
Minhyo Jeong
(snucsl.ta@gmail.com)

Systems Software &
Architecture Lab.

Seoul National University

Spring 2022

4190.103A-001: Programming Practice Lab. 6



실습조

실습조

- @shawn950303
 - 1조: ‘ㄱ’ ~ ‘ㅈ’ 사이에 해당하는 성을 가진 학생
- @euidong
 - 2조: ‘ㅇ’ ~ ‘ㅎ’ 사이에 해당하는 성을 가진 학생

과제 풀이

Lab 5 과제 1

Lab 5 - 과제1

- 반쪽짜리 크리스마스 트리를 만들어봅시다
- 반쪽짜리 크리스마스 트리는 다음과 같은 조건을 만족합니다
 - 트리의 모양은 '*' 문자를 이용하여 표현합니다
 - 왼쪽으로 정렬된 직각삼각형이 순차대로 쌓여있는 모습입니다
 - 1 층은 '*'이 1개
 - 2 층은 삼각형의 가로, 높이가 2인 직각삼각형
 - 3 층은 삼각형의 가로, 높이가 3인 직각삼각형
 - ...

Lab 5 - 과제1

- 출력 예

```
> make -s
> ./main
1
*
> █
```

```
> make -s
> ./main
3
*
*
**
*
**
***
> █
```

```
> make -s
> ./main
5
*
*
**
*
**
***
*
**
***
****
*
**
***
****
*****
> █
```

Lab 5 - 과제1

- 다음 조건을 만족할 수 있도록 프로그래밍 해보세요
- 사용자에게 트리 층의 개수를 입력 받습니다
- 사용자는 올바른 값만 입력한다고 가정합니다
 - 1 이상의 자연수
 - 입력 범위: 1 ~ 10
- 입력 받은 만큼 층을 갖는 반쪽짜리 크리스마스 트리를 그려보세요!

Lab 5 - 과제1

- 첫번째 for 문은 전체 트리 중에서 직각삼각형의 개수를 결정합니다.
- num이 3인 경우 i는 0, 1, 2의 값을 가집니다.
- 3이 입력인 경우 다음과 같이 3번의 loop를 통해 트리를 만듭니다.

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

```
int main(void)
```

```
{
```

```
    int num;
```

```
    scanf("%d", &num);
```

```
    for (int i = 0; i < num; i++)
```

```
    {
```

```
        for (int j = 0; j < i + 1; j++)
```

```
        {
```

```
            for (int k = 0; k < j + 1; k++)
```

```
            {
```

```
                printf("*");
```

```
            }
```

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

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

```
➤ make -s
```

```
➤ ./main
```

```
3
```

```
*
```

```
*
```

```
**
```

```
*
```

```
**
```

```
***
```

```
➤
```

Lab 5 - 과제1

- 두번째 for 문은 각각의 직각삼각형의 높이를 결정합니다.
- 앞의 예제에서 계속 하자면, i 는 0, 1, 2의 범위를 가지고 있고 이에 따라 두번째 루프의 회전수가 $j < i + 1$ 조건문에 따라 정해집니다.

```
...   for (int j = 0; j < i + 1; j++)  
      {  
          for (int k = 0; k < j + 1; k++)  
          {  
              printf("*");  
          }  
          printf("\n");  
      }  
  }
```

```
➤ make -s  
➤ ./main  
3  
*  
*  
**  
*  
**  
***  
➤
```

Lab 5 - 과제1

- 세번째 for 문은 각 직각삼각형의 너비를 결정합니다.
- 앞서 $i \rightarrow j$ 에서 이어지는 범위 지정에 따라 숫자만큼 별을 출력합니다.

```
...   for (int j = 0; j < i + 1; j++)  
    {  
        for (int k = 0; k < j + 1; k++)  
        {  
            printf("*");  
        }  
        printf("\n");  
    }  
}
```

```
➤ make -s  
➤ ./main  
3  
*  
*  
**  
*  
**  
***  
➤
```

Lab 5 - 과제1 학생답안

```
int main(void) {
    int num;
    int N = 1;

    scanf("%d", &num);

    while(N < num+1){
        //int N = 1;
        for(int i = 0; i < N ; i++ ){
            for(int j = 0; j < i+1; j++){
                printf("*");
            }
            printf("\n");
        }
        N++;
        //if (N==num) break;
    }
    return 0;
}
```

Lab 5 - 과제1 학생답안

```
int main(void)
{
    int num;
    scanf("%d", &num);
    for (int k = 0; k < num; k++)
    { // k가 num이 될 때까지 아래 두 개의 for문 반복
        for (int i = 0; i < k + 1; i++)
        { // k줄마다 i개의 \n 출력
            for (int j = 0; j < i + 1; j++)
            { // i줄마다 j개의 *출력
                printf("*");
            }
            printf("\n");
        }
    }
    return 0;
}
```

Lab 5 과제 2

Lab5 - 과제2

- 다음 문제를 해결할 수 있는 프로그램을 만들어보세요!
- 사용자로부터 0~99 사이의 정수를 하나 입력 받습니다
 - 사용자는 항상 옳은 값만 입력한다고 가정합니다
- 주어진 규칙을 통해서 입력 받은 정수를 계산합니다
- 계산된 숫자가 원래의 입력 받은 숫자로 돌아오는데 걸린 사이클의 수를 구해보세요

Lab5 - 과제2

- 주어진 수의 십의 자리와 일의 자리를 더합니다
 - 주어진 수가 한 자리 수 라면, 10을 곱해서 2자리로 만든 후 계산합니다
 - 0을 입력 받았다면, 10을 곱해도 0이므로 0과 0을 더하는 과정이 됩니다
- 주어진 수의 1의 자리와, 더해서 나온 수의 1의 자리를 조합합니다

입력!

7 **6**



7 + 6 = **13**

결과!

6 **3**



Lab5 - 과제2

- 계산 과정
 - 이해를 위한 출력 결과입니다
 - 과제는 총 사이클 수만 출력하시면 됩니다

```
➤ make -s
➤ ./main
76
Cycle 1: 63
Cycle 2: 39
Cycle 3: 92
Cycle 4: 21
Cycle 5: 13
Cycle 6: 34
Cycle 7: 47
Cycle 8: 71
Cycle 9: 18
Cycle 10: 89
Cycle 11: 97
Cycle 12: 76
12
➤ □
```

Lab5 - 과제2

- 새롭게 만들어진 숫자를 이용하여 같은 계산을 반복합니다
- 처음에 입력 받은 숫자가 다시 나올 때 까지, 이 계산을 몇 번 반복해야하는지 알아내는 프로그램을 작성해보세요

```
➤ make -s
➤ ./main
76
12
➤ █
```

```
➤ make -s
➤ ./main
0
1
➤ █
```

```
➤ make -s
➤ ./main
33
60
➤ █
```

```
➤ make -s
➤ ./main
24
20
➤ █
```

Lab5 - 과제2 - 풀이

- 전체 코드 입니다.

```
#include <stdio.h>
int main(void) {
    int num, temp;
    int sum;
    int count;

    scanf("%d", &num);
    temp = num;

    do {
        sum = temp / 10 + temp % 10;
        temp = (temp % 10) * 10 + (sum % 10);
        count++;
    } while (temp != num);

    printf("%d\n", count);
}
```

Lab5 - 과제2 - 풀이

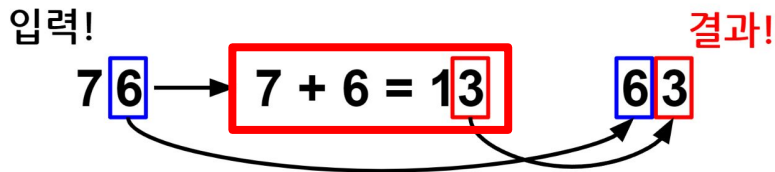
- 초기화를 수행합니다.
- 들어온 값은 비교를 위해 num에 남겨두고
- 연산을 위해 temp 변수를 추가로 만들어주었습니다.
- 몇 번 loop을 돌았는지 확인하기 위해 count 변수를 사용합니다.

```
#include <stdio.h>
int main(void) {
    int num, temp;
    int sum;
    int count;

    scanf("%d", &num);
    temp = num;
```

Lab5 - 과제2 - 풀이

- 입력이 10~99사이로 고정되어 있으므로
- 각 자리 수의 합을 나눗셈과 modular연산으로 구합니다.
- 예제에서 $76/10 + 76\%10$ 연산의 결과로 sum에는 13이 저장 됩니다.



```
#include <stdio.h>

int main(void) {
    int num, temp;
    int sum;
    int count;

    scanf("%d", &num);
    temp = num;

    do {
        sum = temp / 10 + temp % 10;
        temp = (temp % 10) * 10 + (sum % 10);
        count++;
    } while (temp != num);

    printf("%d\n", count);
}
```

Lab5 - 과제2 - 풀이

- sum에는 13이 저장되어 있고,
- temp에는 76이 저장되어 있습니다.
- $(temp \% 10) * 10$ 의 연산으로 60을 만들고,
- $13 \% 10$ 의 연산으로 3이라는 값의 합을 temp에 저장합니다.

입력!

76 → 7 + 6 = 13

결과!

63

```
do {  
    sum = temp / 10 + temp % 10;  
    temp = (temp % 10) * 10 + (sum % 10);  
    count++;  
} while (temp != num);  
  
printf("%d\n", count);  
}
```

Lab5 - 과제2 - 풀이

- temp가 입력된 숫자와
- 같지 않으면 다시 위 로직을 수행하고,
- 같으면 loop를 종료합니다.

```
do {  
    sum = temp / 10 + temp % 10;  
    temp = (temp % 10) * 10 + (sum % 10);  
    count++;  
} while (temp != num);  
  
printf("%d\n", count);  
}
```

Lab 5 - 과제2 학생답안

```
int main(void)
{
    int n;
    int cnt = 0;
    int a, b;
    scanf("%d", &n);
    a = n;
```

```
    while (1)
    {
        cnt++;
        if (a / 10 == 0)
        {
            b = a;
            a = (a % 10) * 10 + (b % 10);
        }
        else
        {
            b = (a / 10) + (a % 10);
            a = (a % 10) * 10 + (b % 10);
        }
        // printf("Cycle %d: %d\n", cnt, a);
        if (a == n)
            break;
    }
    printf("%d\n", cnt);
    return 0;
}
```


Lab 5 - 과제2 학생답안

```
int main(void)
{
    int n;
    scanf("%d", &n);
    int save = n;
    int b;

    int dap = 1;
    b = (n / 10) + (n % 10);
    b %= 10;
    b += (n % 10) * 10;
    n = b;
```

```
while (n != save && n != 0)
{
    b = n / 10 + n % 10;
    b %= 10;
    b += (n % 10) * 10;
    n = b;
    dap++;
}

printf("%d\n", dap);
}
```

실습 1

실습 1 - 설명

- 두 개의 정수를 Parameter로 입력 받고, 두 수를 더한 결과를 리턴하는 함수를 작성하시오.

```
➤ make -s
➤ ./main
1 2
3
➤ ./main
10 20
30
➤ ./main
345 456
801
➤ █
```

실습 1 - 설명

- 두 개의 정수를 Parameter로 입력 받고, 두 수를 더한 결과를 리턴하는 함수를 작성하시오.
 - 두 개의 정수 변수를 선언하고 입력을 받습니다.

```
#include <stdio.h>

int add(int a, int b)
{
    return a+b;
}

int main(void) {
    int a, b;
    scanf("%d %d", &a, &b);
    int sum = add(a,b);
    printf("%d\n", sum);
    return 0;
}
```

실습 1 - 설명

- `int add(int a, int b);`라는 함수를 만들었습니다.
- 입력받은 `a`, `b`를 `add` 함수로 전달합니다.

```
#include <stdio.h>

int add(int a, int b)
{
    return a+b;
}

int main(void) {
    int a, b;
    scanf("%d %d", &a, &b);
    int sum = add(a,b);
    printf("%d\n", sum);
    return 0;
}
```

실습 1 - 설명

- add 함수가 실행되고
- a+b로 계산된 결과가 return 됩니다.
- 그리고 이는 sum에 저장됩니다.

```
#include <stdio.h>

int add(int a, int b)
{
    return a+b;
}

int main(void) {
    int a, b;
    scanf("%d %d", &a, &b);
    int sum = add(a,b);
    printf("%d\n", sum);
    return 0;
}
```

실습 1 - 설명

- 두 개의 양의 정수를 입력 받고, 두 수를 덧셈, 뺄셈, 곱셈, 나눗셈하는 함수를 각각 작성하시오.
 - 입력: 양의 정수 N M
 - 단, 큰 수에서 작은 수를 빼고, 나누도록 합니다.
 - 따라서 N이 M보다 작은 경우 두 변수의 값을 swap 합니다.
 - add() 함수를 응용하여 아래 함수들도 만들어 봅시다.
 - sub(int a, int b)
 - mul(int a, int b)
 - div(int a, int b)
 - 그리고 그 결과도 출력해봅시다.

```
❏ make -s
❏ ./main
1 2
3
1
2
2
❏ ./main
2 1
3
1
2
2
```

실습 1 - 설명

- add와 동일한 방식으로 함수를 만들어 보았습니다.

```
int sub(int a, int b) {
    return a-b;
}

int mul(int a, int b){
    return a*b;
}

int div(int a, int b){
    return a/b;
}

void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

```
int main(void) {
    int a, b;
    scanf("%d %d", &a, &b);
    if (b>a)
        swap(&a, &b);

    printf("%d\n", add(a,b));
    printf("%d\n", sub(a,b));
    printf("%d\n", mul(a,b));
    printf("%d\n", div(a,b));
    return 0;
}
```


실습 1 - 제출

- 매개 변수로 정수 2개를 넘겨 받아 두 수 사이에 있는 정수의 합을 리턴하는 함수를 작성하고, 이를 지정된 형식으로 출력하시오.
 - 입력: 정수 두 개 [0, 100]
 - 출력: 두 수 사이에 있는 정수의 합

```
➤ make -s
➤ ./main
1 3
2
➤ ./main
1 5
9
➤ ./main
50 60
495
➤
```

실습 2

실습 2 - 설명

- 별을 이용하여 다양한 그림을 그려봅시다! (계속)
- printf와 '*' 문자를 활용해 봅시다.
- 반복문에 조금 더 익숙해질 수 있습니다.

실습 2

- for문을 이용하여 뒤집어진 직각삼각형을 출력해봅시다.

```
./main
1
*
./main
3
***
**
*
./main
5
*****
****
***
**
*
```

실습 2 - 설명

- 뒤집어진 직각삼각형을 출력해봅시다.

```
#include <stdio.h>
int main(void)
{
    int i, j, num;
    scanf("%d", &num);

    for ( i = 0; i < num; i++ )
    {
        for ( j = 0; j < num-i; j++ )
            printf("*");
        printf("\n");
    }
    return 0;
}
```

```
./main
1
*
./main
3
***
**
*
./main
5
*****
****
***
**
*
```

실습 2

- 숫자를 입력받아, 해당 숫자 만큼 밑변의 너비를 가지는 피라미드 모양을 출력해봅시다.

```
> make -s
> ./main
1
*
> ./main
3
*
***
> ./main
5
*
***
*****
> ./main
7
*
***
*****
*****
```

```
> ./main
15
*
***
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

실습 2 - 설명

- 코드는 다음과 같습니다.

```
#include <stdio.h>
int main(void) {
    int s;
    scanf("%d", &s);
    for (int i = 0; i < s; i+=2) {
        for (int k = 0; k < (s - i) / 2; k++) {
            printf(" ");
        }
        for (int j = 0; j <= i; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

실습 2 - 제출

- 숫자를 입력받아, 해당 숫자 만큼 높이를 가지는 다이아몬드 모양을 출력해봅시다. (단 입력은 홀수 정수만 가능하다. 범위: [1, 15])

```
➤ ./main
1
*
➤ ./main
7
*
***
*****
*****
*****
***
*
```

```
➤ ./main
15
*
***
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
***
*
```


실습 3

실습 3 - 설명

- 정수를 십진법, 십육진법, 팔진법으로 출력하기
 - 하나의 정수를 입력받아서 아래와 같이 출력해봅시다.
 - 같은 숫자를 손쉽게 여러 진법으로 표현할 수 있습니다.

```
#include <stdio.h>

int main(void)
{
    int n;
    scanf("%d", &n);
    printf("%d %x %o\n", n, n, n);
    return 0;
}
```

```
❖ make -s
❖ ./main
10
❖ ./main
10 a 12
❖ ./main
20
❖ ./main
20 14 24
❖ ./main
16
❖ ./main
16 10 20
❖ ./main
9
❖ ./main
9 9 11
❖
```

실습 3 - 설명

- 정수를 다양하게 표현해봅시다.
 - 정수 서식 지정자 %d에 폭을 지정해 보겠습니다.
 - %와 d 사이에 숫자를 넣으면 출력 폭을 지정할 수 있습니다.

```
#include <stdio.h>

int main(void)
{
    int n;
    scanf("%d", &n);
    printf("%4d\n", n); // 출력 폭을 4칸으로 지정
    printf("%6d\n", n); // 출력폭을 6칸으로 지정
    return 0;
}
```

```
➤ make -s
➤ ./main
1111
1111
   1111
➤ ./main
1
   1
      1
➤ ./main
20
   20
      20
➤ █
```

실습 3 - 설명

- 정수를 다양하게 표현해봅시다.
 - 형식은 다음과 같습니다: %[플래그][폭][.정밀도][길이]서식지정자
 - 플래그를 사용해서 아까 빈 공간에 0을 출력해봅시다.

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    int n;
    scanf("%d", &n);
    printf("%04d\n", n); // 출력 폭을 4칸. 빈공간 0.
    printf("%06d\n", n); // 출력폭을 6칸. 빈공간 0.
    return 0;
}
```

```
❯ make -s
❯ ./main
1111
1111
001111
❯ ./main
1
0001
000001
❯ ./main
20
0020
000020
❯
```

실습 3 - 설명

- 실수를 소수점 아래 자릿수 설정 하기
 - 한 개의 실수를 입력받아 정밀도를 서로 다르게 출력해봅시다.
 - %.1f 와 같이 % 뒤에 . 숫자로 조절할 수 있습니다.

```
#include <stdio.h>
#include <math.h>
int main(void)
{
    float n;
    scanf("%f", &n);
    printf("%f\n", n);
    printf("%.1f\n", n);
    printf("%.2f\n", n);
    printf("%.3f\n", n);
    printf("%.4f\n", n);
    printf("%.5f\n", n);
    return 0;
}
```

```
➤ make -s
➤ ./main
1.1111
1.111100
1.1
1.11
1.111
1.1111
1.11110
➤
```

실습 3 - 제출

- RGB 컬러 출력

- 세개의 십진수 정수를 입력받아 각 컬러별로 십육진법의 지정된 형식으로 출력하기
- 입력: 세 개의 십진수 정수. 입력 범위는 각각 [0,255].
 - 각 정수는 Red, Green, Blue 값에 대응한다.
- 출력: #RRGGBB 형식
 - RR: 첫번째 입력값을 16진수로 변경한 값 (e.g. 255->FF)
 - GG: 두번째 입력값을 16진수로 변경한 값
 - BB: 세번째 입력값을 16진수로 변경한 값
 - 7글자 사이에 공백 없이 출력할 것
 - 알파벳은 대문자로 출력할 것

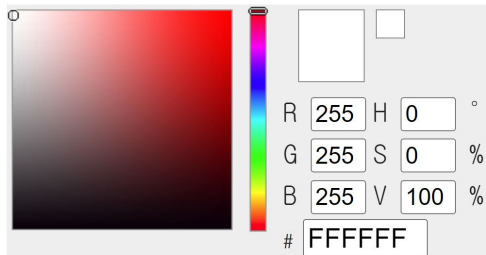
```
❯ make -s
❯ ./main
255 255 255
#FFFFFF
❯ ./main
0 0 0
#000000
❯ ./main
10 20 30
#0A141E
❯ ./main
40 50 60
#28323C
❯ ./main
123 45 87
#7B2D57
❯
```

실습 3 - 제출

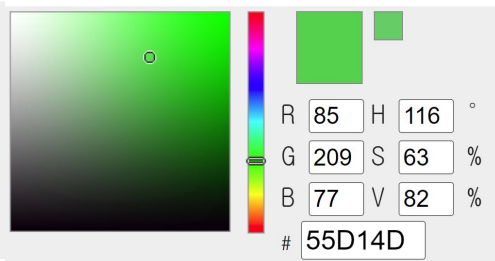
- RGB 컬러 출력

- 참고 사이트: https://www.rapidtables.com/web/color/RGB_Color.html

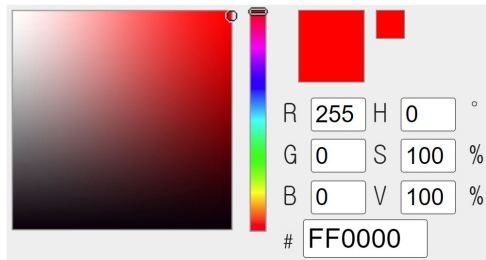
- RGB color picker



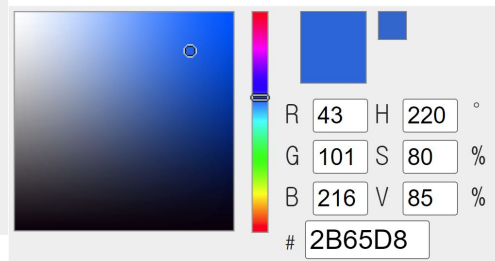
- RGB color picker



- RGB color picker



- RGB color picker



실습 4

실습 4 - 설명

- math library의 `cos()`, `sin()` 함수를 사용해 봅시다.
 - `rand()` 함수의 경우 괄호 안에 아무 것도 입력하지 않아도 결과값이 출력 되었습니다.
 - `cos()`, `sin()`의 경우에는 괄호 안에 각도를 넣으면 해당하는 삼각함수 값으로 리턴 됩니다.

실습 4 - 설명

- math library의 `cos()`, `sin()` 함수를 사용해 봅시다.
 - 아래 코드를 참고해 30° 45° 90° 의 코사인, 사인 값을 출력해봅시다.
 - 이를 위해, `<math.h>` library를 추가로 include 해줍니다.
 - 아래 예제와 같이 `cos(theta)`, `sin(theta)`의 결과값을 출력해봅시다.

```
#include <stdio.h>
#include <math.h>

int main()
{
    double theta = 30.0, cos_theta, sin_theta;
    cos_theta = cos(theta);
    sin_theta = sin(theta);

    printf("cos(%lf) => %.2lf\n", theta, cos_theta);
    printf("sin(%lf) => %.2lf\n", theta, sin_theta);
    return 0;
}
```

실습 4 - 설명

- math library의 `cos()`, `sin()` 함수를 사용해 봅시다.
 - 정상적인 값으로 출력되지 않는 것을 볼 수 있습니다.

```
➤ make -s
➤ ./main
cos(30.000000) => 0.15
sin(30.000000) => -0.99
➤
```

x	sin x	cos x	tan x
29.8	0.496973961	0.867765453	0.5727054
29.9	0.49848774	0.866896749	0.575025504
30	0.5	0.866025404	0.577350269
30.1	0.501510737	0.865151421	0.579679725

실습 4 - 설명

- c math library의 삼각함수들은 radian으로 입력을 받습니다.

function

COS

C90	C99	C++98	C++11	?
<pre>double cos (double x); float cosf (float x); long double cosl (long double x);</pre>				

Compute cosine

Returns the cosine of an angle of x radians.

실습 4 - 설명

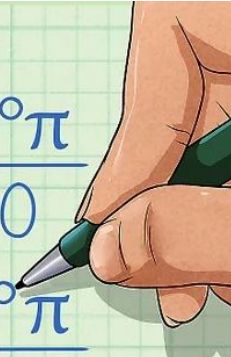
- 아래 공식을 이용해서 degree를 radian으로 변환해보세요.

Example : $1^{\circ} = \frac{\pi}{180}$

$$120^{\circ} \times \frac{\pi}{180}$$
$$30^{\circ} \times \frac{\pi}{180}$$
$$225^{\circ} \times \frac{\pi}{180}$$


wikiHow

Example :

$$120^{\circ} \times \frac{\pi}{180} = \frac{120^{\circ}\pi}{180}$$
$$30^{\circ} \times \frac{\pi}{180} = \frac{30^{\circ}\pi}{180}$$
$$225^{\circ} \times \frac{\pi}{180} = \frac{225^{\circ}\pi}{180}$$


wikiHow

실습 4 - 설명

- 아래 주어진 공식을 활용해서 degree에서 radian으로 변환하는 코드를 작성해보세요.
 - radian 변환 공식은 다음과 같습니다.
 - $x \text{ radian} = (\text{degree} * \text{PI}) / 180$
 - PI의 값은 3.141592654으로 사용하세요.

$$2\pi \text{ rad} = 360^\circ$$

$$1 \text{ rad} = \frac{360^\circ}{2\pi} = \frac{180^\circ}{\pi} \approx 57.29577951^\circ$$

실습 4 - 설명

- 아래 주어진 공식을 활용해서 degree에서 radian으로 변환하는 코드를 작성해보세요.
 - radian 변환 공식은 다음과 같습니다.
 - $x \text{ radian} = (\text{degree} * \text{PI}) / 180$
 - PI의 값은 3.141592654으로 사용하세요.
 - ※ 아래 코드를 함수 형태로도 작성해보세요!

```
#define PI 3.141592654

double deg = 30.0;
double rad = deg * PI / 180.0;
```

실습 4 - 설명

- 입력을 radian으로 변환해서 아까 값을 다시 출력해봅시다.
 - 30° 45° 90°를 radian 값으로 변환시키고 다시 sin, cos 값을 구해봅시다

```
➤ make -s
➤ ./main
sin(0.52) => 0.50
cos(0.52) => 0.87
➤
```

x	sin x	cos x	tan x
29.8	0.496973961	0.8677765453	0.5727054
29.9	0.49848774	0.866896749	0.575025504
30	0.5	0.866025404	0.577350269
30.1	0.501510737	0.865151421	0.579679725

실습 4 - 설명

```
#include <stdio.h>
#include <math.h>
#define PI 3.141592654

int main()
{
    double theta = 30.0, cos_theta, sin_theta;
    theta = theta*PI/180.0;
    cos_theta = cos(theta);
    sin_theta = sin(theta);

    printf("sin(%.21f) => %.21f\n", theta, sin_theta);
    printf("cos(%.21f) => %.21f\n", theta, cos_theta);

    return 0;
}
```

```
➤ make -s
➤ ./main
sin(0.52) => 0.50
cos(0.52) => 0.87
➤
```

실습 4

- 2개의 양의 정수를 n , m 을 입력 받아 n degree 부터 $n+m$ degree까지 1도 간격으로 $\tan$ 값을 출력하시오.
 - 입력 : n , m ($0 \leq n \leq 45$, $1 \leq m \leq 3$)
 - ※ 단, 정상적인 입력만 가정한다.
 - 출력: 조건에 맞는 탄젠트 값 (※ 우측의 예시 참조)

```
➤ make -s
➤ ./main
45 1
1.000000 1.035530
➤ ./main
45 2
1.000000 1.035530 1.072369
➤ ./main
45 3
1.000000 1.035530 1.072369 1.110613
➤ ./main
15 1
0.267949 0.286745
➤ ./main
15 2
0.267949 0.286745 0.305731
➤ ./main
15 3
0.267949 0.286745 0.305731 0.324920
➤
```

실습 4 - 설명

- 2개의 정수를 n, m을 입력 받아 n degree 부터 n+m degree까지 1도 간격으로 tan 값을 출력하는 프로그램을 작성해봅시다.

```
#include <stdio.h>
#include <math.h>

#define PI 3.141592654

double deg2rad(double deg)
{
    return /* write your code */
}

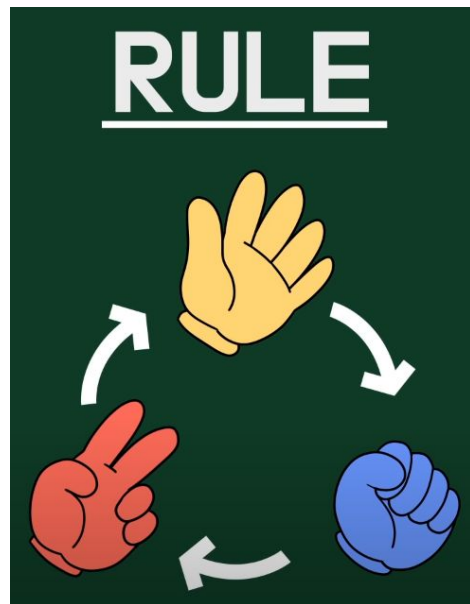
int main(void)
{
    int n, m;
    scanf("%d %d", &n, &m);

    for(int i = n; i <= n+m; i++)
    {
        printf("%f ", tan(deg2rad((double)i)));
    }
    printf("\n");
    return 0;
}
```

실습 5

실습 5 - 설명

- 가위 바위 보
 - 컴퓨터와 하는 가위 바위 보 게임을 만드세요.
 - 사용자 입력
 - ['X', 'O', '#'] - 각각 가위 바위 보에 대응합니다.
 - ['!'] - 게임을 종료합니다.
 - 우선, 비정상적인 입력은 고려하지 않습니다.
 - 컴퓨터의 값은 time을 seed로 이용한 random number 입니다.
 - 사용자가 종료할 때까지 계속 실행됩니다.
 - 출력 - 다음장 예시 참고
 - 컴퓨터와 사용자의 패에 맞는 승부 결과
 - 컴퓨터의 패 - 사용자 입력과 같은 형식



실습 5 - 설명

● 가위 바위 보

- 다음과 같이 출력하시면 됩니다.
- 조건을 만족하며 출력형식을 자유롭게 작성해 보세요.

```
➤ make -s
➤ ./main
```

```
*****
* 가위 ('X'), 바위 ('O'), 보 ('#')?
* 종료하려면 '!'을 입력하세요.
*****
* 입력: █
```

```
Q x ➤ make -s  
➤ ./main
```

```
*****
* 가위('X'), 바위('O'), 보('#')?
* 종료하려면 '!'을 입력하세요.
*****
* 입력 : 0
* User loses. (computer: #)
*****
* 가위('X'), 바위('O'), 보('#')?
* 종료하려면 '!'을 입력하세요.
*****
* 입력 : █
```

```

> make -s
> ./main
*****
* 가위 ('X'), 바위 ('O'), 보 ('#')?
* 종료하려면 '!'을 입력하세요.
*****
* 입력: 0
* User wins. (computer: X)
*****
* 가위 ('X'), 바위 ('O'), 보 ('#')?
* 종료하려면 '!'을 입력하세요.
*****
* 입력: X
* User loses. (computer: O)
*****
* 가위 ('X'), 바위 ('O'), 보 ('#')?
* 종료하려면 '!'을 입력하세요.
*****
* 입력: #
* User wins. (computer: O)
*****
* 가위 ('X'), 바위 ('O'), 보 ('#')?
* 종료하려면 '!'을 입력하세요.
*****
* 입력: !

```

실습 5 - 설명

- 가위 바위 보
 - 우선 게임을 위한 전체적인 구조를 짭니다.
 - 랜덤 숫자 생성과, 반복문,
 - 그리고 종료 조건을 우선 처리하도록 시작했습니다.

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

int main()
{
    srand(time(NULL));
    while (1) //무한 반복
    {
        int com = rand() % 3; // 0, 1, 2
        char user;
        scanf(" %c", &user); // 'X' 'O' '#'
        if (user == '!')
            break; // Loop 벗어남
    }

    return 0;
}
```

실습 5 - 설명

- 가위 바위 보
 - User 입력값과, 컴퓨터의 결과값을 가지고 있으니,
 - 'X' 'O' '#'과 같은 정상적인 입력에 대해서 처리할 수 있습니다.

```
if (user == '!')
    break;
else if (user == 'O' || user == 'X' || user == '#')
{
    int decoded = code2num(user);
    printResult(game(com, decoded));
    printf("(computer: %c)\n", num2code(com));
}
```


실습 5 - 설명

- 가위 바위 보

- int code2num(char a) 함수는 입력으로 받은 가위바위보 코드(X, O, #)을 숫자로 바꿔주는 코드입니다.
- 정수로 랜덤 생성된 컴퓨터의 값과 비교하기 위한 사전 작업입니다.

```
if (user == '!')
    break;
else if (user == 'O' || user == 'X' || user == '#')
{
    int decoded = code2num(user);
}
```

```
int code2num(char a)
{
    if (a == 'X')
        return 0; // scissors
    else if (a == 'O')
        return 1; // rock
    else if (a == '#')
        return 2; // paper
    else
        printf("wrong input.\n");
}
```

실습 5 - 설명

- 가위 바위 보

- int code2num(char a) 함수는 입력으로 받은 가위바위보 코드(X, O, #)을 숫자로 바꿔주는 코드입니다.
- 정수로 랜덤 생성된 컴퓨터의 값과 비교하기 위한 사전 작업입니다.

```
if (user == '!')
    break;
else if (user == 'O' || user == 'X' || user == '#')
{
    int decoded = code2num(user);
}
```

```
int code2num(char a)
{
    if (a == 'X')
        return 0; // scissors
    else if (a == 'O')
        return 1; // rock
    else if (a == '#')
        return 2; // paper
    else
        printf("wrong input.\n");
}
```

실습 5 - 설명

- 가위 바위 보

- game() 함수를 통해서 게임의 승패를 가립니다.
- 편의상 draw, win, lose를 상수 변수를 사용했습니다.
- 변수 앞에 const 가 붙는 경우 초기화 값 이외에 변경은 불가능합니다.
- 가위바위보 조건에 따라 승/패/무승부를 return 합니다.

```
else if (user == 'O' || user == 'X' || user == '#')  
{  
    int decoded = code2num(user);  
    game(com, decoded)  
}
```

```
int game(int comp, int user)  
{ // draw 0 win 1 lose 2  
    const int draw = 0;  
    const int win = 1;  
    const int lose = 2;  
  
    if (comp == user)  
        return draw;  
  
    if (user == 0 && comp == 2) // u: s, c: p  
        return win;  
    else if (user == 0 && comp == 1) // u: s, c: r  
        return lose;  
    else if (user == 1 && comp == 0) // u: r, c: s  
        return win;  
    else if (user == 1 && comp == 2) // u: r, c: p  
        return lose;  
    else if (user == 2 && comp == 1) // u: p, c: r  
        return win;  
    else if (user == 2 && comp == 0) // u: p, c: s  
        return lose;  
}
```

실습 5 - 설명

- 가위 바위 보
 - game() 함수를 통해 정해진 결과를 출력하는 함수를 사용하였습니다.
 - 해당 결과값을 출력하는 함수를 작성하였습니다.

```
else if (user == 'O' || user == 'X' || user == '#')
{
    int decoded = code2num(user);
    int result = game(com, decoded);
    printResult(result);
}
```

```
int printResult(int result)
{
    if (result == 0)
        printf("* Draw Game. ");
    else if (result == 1)
        printf("* User wins. ");
    else if (result == 2)
        printf("* User loses. ");
}
```

실습 5 - 설명

- 가위 바위 보
 - 컴퓨터의 가위바위보 값도 출력해주도록 합니다.

```
else if (user == 'O' || user == 'X' || user == '#')
{
    int decoded = code2num(user);
    int result = game(com, decoded);
    printResult(result);
    printf("(computer: %c)\n", num2code(com));
}
```

```
char num2code(int a)
{
    if (a == 0)
        return 'X';
    else if (a == 1)
        return 'O';
    else if (a == 2)
        return '#';
    else
        printf("wrong input.");
}
```

실습 5 - 설명

- 가위 바위 보
 - 필요한 printf 문을 더하여 게임을 완성해보세요.

실습 5 – Optional

- 가위 바위 보
 - 총 N 번만 돌아가는 가위바위보 게임을 작성해보세요.
 - 그리고 몇 번 유저가 이기고 지는지를 기록해서 출력해보세요.
 - 더 좋은 랜덤 형식을 고민해보세요.

과제 1

과제 1 - 문제

- 지구 상에서 두 지점 거리 구하기
 - 앞서 설명한 math library를 이용하면 지구상 두 지점의 거리를 구할 수 있습니다.
 - 아래 “구면 코사인 법칙”을 이용해서 지구상의 두 지점간 거리를 구해보세요.

Radian 각도로 표현된 두 지점 사이 거리

$(lat1, lon1), \quad (lat2, lon2)$

$$\cos \theta = \sin(lat1)\sin(lat2) + \cos(lat1)\cos(lat2)\cos(lon2 - lon1)$$

$$\theta = \arccos(\cos(\theta))$$

$$distance = r\theta$$

과제 1 - 문제

- 지구 상에서 두 지점 거리 구하기
 - math library의 $\cos()$, $\sin()$, $\arccos()$ 를 사용합니다.
 - 지구의 반지름(r)은 6371 km를 사용합니다.
 - radian 변환에 필요한 파이 값은 3.14159265358979323846 을 사용합니다.

Radian 각도로 표현된 두 지점 사이 거리

$(lat1, lon1), \quad (lat2, lon2)$

$$\cos \theta = \sin(lat1)\sin(lat2) + \cos(lat1)\cos(lat2)\cos(lon2 - lon1)$$

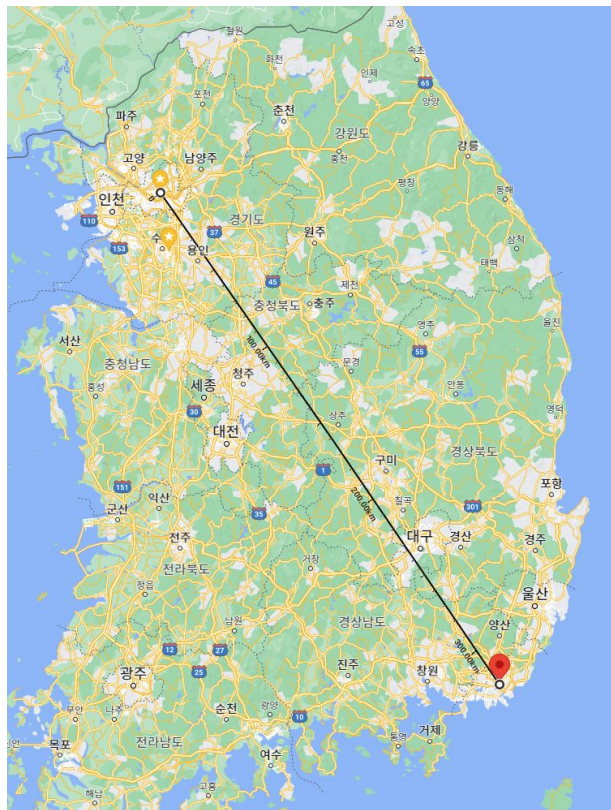
$$\theta = \arccos(\cos(\theta))$$

$$distance = r\theta$$

과제 1 - 문제

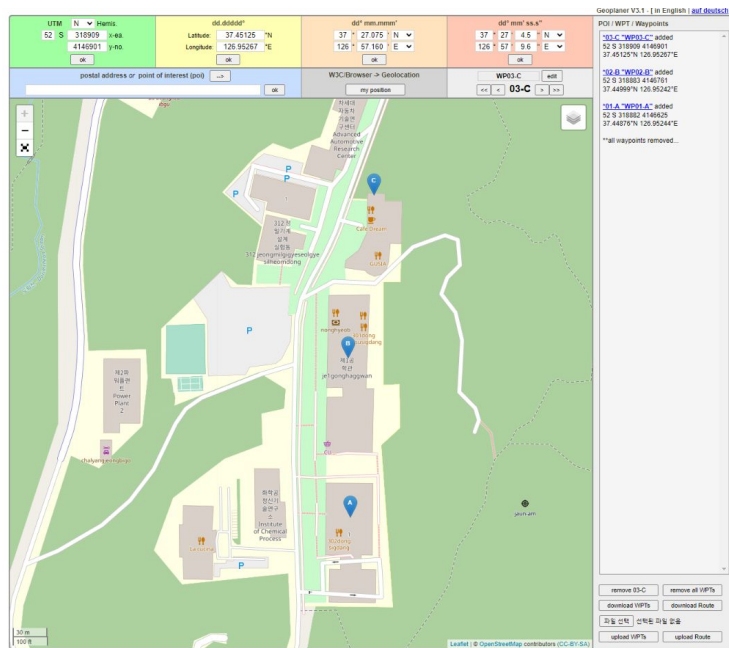
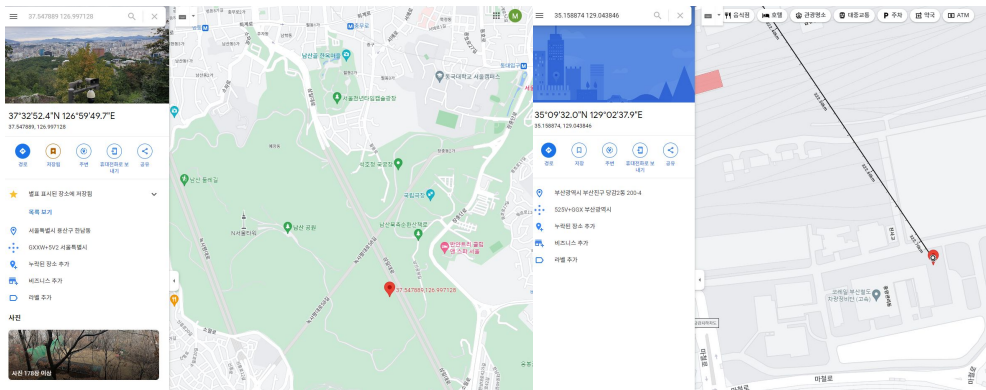
- 지구 상에서 두 지점 거리 구하기
 - 입력 예시:
 - 첫 번째 위도/경도 : 37.547889 / 126.997128
 - 두 번째 위도/경도: 35.158874 / 129.043846
 - 출력 예시: 322.7
 - 단위 km, 소수점 이하 1자리까지만 출력할 것

```
./main
37.547889 126.997128 35.158874 129.043846
322.7
```



과제 1 - 문제

- 스스로 테스트 데이터 만들기
 - google maps 에서 지도를 우클릭하면 해당 지점의 위도와 경도를 알 수 있습니다.
 - 다른 지도 서비스도 활용해보세요 (e.g. <https://www.geoplaner.com/>)



과제 1 - 평가 기준

- 기준 1
 - 프로그램이 동작하는가? - 1점
- 기준 2
 - 정상적으로 4개의 실수를 입력 받는가? - 1점
- 기준 3
 - 정해진 형식대로 결과를 출력하고 계산이 정확한가? - 3점

과제 1 – Tips

- 아래 template code의 function prototype을 활용해서 코드를 작성해보세요.

```
#include <math.h>
#define pi 3.14159265358979323846
#define radius 6371

/* complete functions */
double deg2rad(double deg);
double distance(double lat1, double lon1, double lat2, double lon2);

int main(void)
{
    /* write your code */
}
```

과제 2

과제 2 - 문제

- 확장된 소수 판별 프로그램

- 두 양의 정수 N, M을 입력 받아 N 보다 큰 소수를 작은 것부터 차례로 M개 출력하는 프로그램을 작성하시오
- 입력: 두 개의 양의 정수 N, M ($1 \leq N \leq 100$, $1 \leq M \leq 5$)
 - 예시: 31 5
- 출력: N보다 큰 M개의 소수 - 각 숫자 사이는 space로 띄운다. 맨 뒤도 띄우고 개행함.
 - 예시: 37 41 43 47 53
- 예전에 for, modular 연산으로 소수 판별을 연습 했습니다.
- ※ 1과 그 수 자신 이외의 자연수로는 나눌 수 없는 자연수

```
❏ make -s
❏ ./main
31 5
37 41 43 47 53
❏
```


과제 2 - 평가 기준

- 기준 1
 - 프로그램이 동작하는가? - 1점
- 기준 2
 - 정상적으로 2개의 숫자를 입력 받는가? - 1점
- 기준 3
 - 정해진 형식대로 결과를 출력하고 계산이 정확한가? - 3점

과제 2 – Tips

- 소수를 판별하는 함수를 작성해보세요.
- 그리고 여러가지 소수를 빠르게 판별하는 기법들을 사용해보세요.