



R E P O R T

제8장 함수

과목명	C 프로그래밍 I
분반	4 분반
교수	문 일 영
학번	2020136129
이름	최 수 연
제출일	2020년 6월 14일 일요일

01. 함수에 대한 설명 중 잘못된 것은? (1)

02. 다음 함수 원형 정의 중에서 int형을 반환하는 함수는? (3)

03. 다음의 함수 원형 정의 중에서 잘못된 것은? (1, 2, 4)

04. `int func(int);`의 원형을 가지는 함수의 호출로 옳은 것은? (1)

05. 다음 라이브러리 함수 호출의 반환값을 쓰라.

(a) `fabs(-1.72)` = -1.72 (b) `floor(1.72)` = 1 (c) `ceil(1.72)` = 2

06. 다음 수식의 반환값의 범위는?

(a) `rand()%10` = 0 ~ 9 (b) `rand()%5+2` = 2 ~ 6

07. 다음의 수학적식을 계산하는 문장을 작성해보자.

(a) $y = \log_{10}x + e^x$

=> `log10(x) + e(x)`

(b) $y = \sin(x) + \sqrt{(x \cdot x - 2 \cdot a) + 2^{10}}$

=> `sin(x) + sqrt(x*x-2*a) + pow(2, 10)`

08. 다음과 같은 기술에 부합하는 함수 원형을 작성하라.

(a) int형 매개 변수 n을 받아서 아무것도 반환하지 않는 `print_error()` 함수

=> `void print_error(int n)`

(b) double형 매개 변수 x, y를 받아서 double형을 반환하는 `larger_of()` 함수

=> `double larger_of(double x, double y)`

(c) 어떠한 매개 변수도 갖지 않고 아무것도 반환하지 않는 `side_effect()` 함수

=> `void side_effect()`

09. 왼쪽 박스의 함수 원형 정의에 적합한 return문을 오른쪽 박스에서 찾아서 서로 연결하라.

`int f(void);` - `return 10+20;`

`void g(int, int);` - `return;`

`double h(double, int);` - `return 1.0;`

10. 다음은 1부터 10까지의 합을 계산하는 프로그램이다. 프로그램의 빈칸을 채워라.

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

```
int f(int n);
```

```
int main(void)
```

```
{
```

```
    printf("%d", f(10));
```

```
    return 0;
```

```
}
```

```
int f(int n)
```

```
{
```

```
    int i, result = 0;
```

```
    for (i = 0; i <= n; i++)
```

```
        result += i;
```

```
    return result;
```

```
}
```

11. 다음의 함수 원형 정의가 올바른지 결정하고 만약 잘못된 점이 있다면 이유를 설명하고 바르게 수정하라.

(d) double f(double x, y);

=> y의 자료형 double이 빠짐. // 답) double f(double x, double y);

(e) (int) f(double x, double y);

=> 반환형 int에 괄호가 있으면 안 됨. // 답) int f(double x, double y);

(f) int f((int)x, (int)y);

=> 매개변수 자료형 int에 괄호가 있으면 안 됨. // 답) int f(int x, int y);

(g) float get_area(radius, pi);

=> 매개변수에 자료형을 입력해야 함. // 답) float get_area(float radius, float pi);

12. 다음의 프로그램에서 붉은 색으로 표시된 부분을 함수로 작성하여 프로그램을 수정하십시오. 함수는 인수와 반환값을 갖도록 설계하라.

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

```
int a(int n);
```

```
int main(void)
```

```
{
```

```
int n;
```

```
printf("정수를 입력하십시오: ");
```

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

```
printf("0부터 %d까지의 합은 %d입니다. \n", n, a(n));
```

```
}
```

```
int a(int n)
```

```
{
```

```
int sum = 0, i;
```

```
for (i = 0; i <= n; i++)
```

```
{
```

```
sum += i;
```

```
}
```

```
return sum;
```

```
}
```

13. 다음 코드에서 잘못된 곳(컴파일 오류 및 경고)을 지적하라. 논리적인 오류도 지적하라.

(a) // int half_of(int x); 에서 세미콜론 제거해야 함.

```
int half_of(int x)
```

```
{
```

```
    return x / 2;
```

```
}
```

(b) // 다음 코드에서 print_message 함수는 매개 변수가 없으므로 3을 제거해야 함.

```
void print_message(void);
```

```
int main(void)
{
    print_message()
}
```

(c) // 함수 선언 시 정의와 매개 변수의 자료형을 하나로 통일해야 함.

```
double half_of(double);
```

```
int main(void)
{
    printf("%f", half_of(10.0));
    return 0;
}
double half_of(double x)
{
    return x / 2.0;
}
```

(d) // sum 값 또는 x+y 값을 return 해야 함.

```
int sum(int x, int y)
{
    sum = x + y;
    return sum;
}
```

(e) // 반환형이 존재하지 않으므로 return을 하면 안 됨.

```
void sum(void)
{
    int x=1, y=2, z=3;
}
```

01. 주어진 실수를 제공하여 반환하는 함수 `double square(double)`을 작성한다.
`square()` 함수를 테스트하는 프로그램을 작성하라.

```
#include <stdio.h>

double square(double);

int main(void)
{
    double x;
    double result;

    printf("정수를 입력하시오 : ");
    scanf("%lf", &x);

    result = square(x);
    printf("주어진 정수 %lf의 제곱은 %lf입니다.\n", x, result);

    return 0;
}

double square(double x)
{
    return x * x;
}
```

Microsoft Visual Studio 디버그 콘솔

정수를 입력하시오 : 2.0
주어진 정수 2.000000의 제곱은 4.000000입니다.

02. 전달된 문자가 알파벳 문자인지 아닌지를 검사하는 함수 `check_alpha()`를 작성하고 이것을 호출하여서 사용자가 입력한 문자가 알파벳('a'에서 'z'까지)인지를 판단하여 출력하는 프로그램을 작성하라.

```
#include <stdio.h>

int check_alpha(char);

int main(void)
{
    char c;
    int result;

    printf("문자를 입력하십시오 : ");
    scanf("%c", &c);

    result = check_alpha(c);

    if (result == 1)
        printf("%c는 알파벳 문자입니다.\n", c);
    else
        printf("%c는 알파벳 문자가 아닙니다.\n", c);

    return 0;
}

int check_alpha(char c)
{
    if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'))
        return 1;
    else
        return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

문자를 입력하십시오 : k
k는 알파벳 문자입니다.

03. 원의 면적을 구하는 문제를 함수로 작성하여 보자. 다음과 같은 식을 이용하라. 원의 면적을 구하는 함수 `cal_area(double radius)`를 작성하고 함수를 호출하여 원의 면적을 출력하는 전체 프로그램을 완성하라.

$$\text{면적} = \text{원주율} * \text{반지름} * \text{반지름}$$

```
#include <stdio.h>
#define PI 3.141592

double cal_area(double radius);

int main(void)
{
    double one;
    double result;
    printf("원의 반지름을 입력하시오 : ");
    scanf("%lf", &one);

    result = cal_area(one);
    printf("원의 면적은 %f입니다.\n", result);

    return 0;
}

double cal_area(double radius)
{
    return PI * radius * radius;
}
```

Microsoft Visual Studio 디버그 콘솔

```
원의 반지름을 입력하시오 : 10.0
원의 면적은 314.159200입니다.
```


04. 우리는 앞에서 윤년을 구하는 알고리즘을 학습하였다. 이것을 함수 `is_leap(int year)` 함수로 작성하고 이 함수를 사용하여 사용자가 입력한 연도가 윤년인지를 출력하는 프로그램을 작성하라.

```
#include <stdio.h>

int is_leap(int year);

int main(void)
{
    int year;
    printf("연도를 입력하시오: ");
    scanf("%d", &year);

    printf("%d년은 %d일입니다.\n", year, is_leap(year));

    return 0;
}

int is_leap(int year)
{
    if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
        return 366;
    else
        return 365;
}
```

Microsoft Visual Studio 디버그 콘솔

연도를 입력하시오: 2012
2012년은 366일입니다.

05. 실수를 정수로 변환하면 소수점 이하는 잘려서 없어지게 된다. 예를 들어서 6.999를 정수로 변환하면 6이 된다. 실수에 0.5를 더하여 소수점 이하를 버리는 반올림 연산을 수행하는 함수 `round(double f)`를 작성하고 테스트하라. 단 `f`는 양수라고 가정하라.

```
#include <stdio.h>

int round(double f);

int main(void)
{
    double num;
    printf("실수를 입력하시오: ");
    scanf("%lf", &num);
    printf("반올림한 값은 %d입니다.\n", round(num));
    return 0;
}

int round(double f)
{
    return (int)(f + 0.5);
}
```

Microsoft Visual Studio 디버그 콘솔

실수를 입력하시오: 3.141592
반올림한 값은 3입니다.

06. 다음과 같은 간단한 기능을 하는 함수들을 작성하고, 사용자로부터 임의의 값을 입력 받은 후에 작성한 함수들을 테스트하여 보자.

(a) 주어진 정수가 짝수이면 1을 반환하고 홀수이면 0을 반환하는 함수 `int even(int n)`

(b) 주어진 정수의 절대값을 구하는 함수 `int absolute(int n)`

(c) 주어진 정수가 음수이면 -1을, 양수이면 1을 0이면 0을 반환하는 함수 `int sign(int n)`

```
#include <stdio.h>

int even(int n);
int absolute(int n);
int sign(int n);

int main(void)
{
    int num;

    printf("정수를 입력하시오: ");
    scanf("%d", &num);

    if (even(num))
        printf("even()의 결과: 짝수\n");
    else
        printf("even()의 결과: 홀수\n");

    printf("absolute()의 결과: %d\n", absolute(num));

    if (sign(num) == 1)
        printf("sign()의 결과: 양수\n");
    else if (sign(num) == -1)
        printf("sign()의 결과: 음수\n");
    else
        printf("sign()의 결과: 0\n");

    return 0;
}

int even(int n)
{
    if (n % 2 == 0)
        return 1;
    else
        return 0;
}

int absolute(int n)
{
    if (n >= 0)
        return n;
    else
        return -n;
}

int sign(int n)
{
    if (n == 0)
        return 0;
    else if (n > 0)
        return 1;
    else
        return -1;
}
```

Microsoft Visual Studio 디버그 콘솔

```
정수를 입력하시오: 12
even()의 결과: 짝수
absolute()의 결과: 12
sign()의 결과: 양수

C:\Users\윤재원\source\repos\...
이 창을 닫으려면 아무 키나 누르십시오.
```

문제가 검색되지 않음

07. 월급에 붙는 소득세를 계산하는 함수 `get_tax(int income)`를 작성하고 테스트하여 보자. 좌표 구간은 1000만 원 이하 8%, 1000만원 초과는 10%로 되어 있다고 가정한다. 사용자로부터 소득을 입력받아서 세금을 계산하는 프로그램을 작성하라.

```
#include <stdio.h>

int get_tax(int income);

int main(void)
{
    int income;
    printf("소득을 입력하시오(만원):");
    scanf("%d", &income);

    printf("소득세는 %d만원입니다.\n", get_tax(income));

    return 0;
}

int get_tax(int income)
{
    if (income <= 1000)
        return (int)(income * 0.08);
    else
        return (int)(1000 * 0.08 + (income - 1000) * 0.1);
}
```

Microsoft Visual Studio 디버그 콘솔

```
소득을 입력하시오(만원):2500
소득세는 230만원입니다.
```

08. sin() 라이브러리 함수를 호출하여서 0도부터 180도까지 10도 단위로 사인 함수 값을 출력하여 보자. 추가적으로 아예 각도를 받아서 사인값을 반환하는 함수 sin_degree(double degree)를 작성하여 문제를 해결할 수 있는가?

```
#include <stdio.h>
#include <math.h>
#define PI 3.141592

double sin_degree(double degree);

int main(void)
{
    int i;

    for (i = 0; i <= 180; i += 10)
    {
        printf("sin(%d)의 값은 %f\n", i, sin_degree(i));
    }

    return 0;
}

double sin_degree(double degree)
{
    return sin((PI * degree)/180.0);
}
```

Microsoft Visual Studio 디버그 콘솔

```
sin(0)의 값은 0.000000
sin(10)의 값은 0.173648
sin(20)의 값은 0.342020
sin(30)의 값은 0.500000
sin(40)의 값은 0.642787
sin(50)의 값은 0.766044
sin(60)의 값은 0.866025
sin(70)의 값은 0.939693
sin(80)의 값은 0.984808
sin(90)의 값은 1.000000
sin(100)의 값은 0.984808
sin(110)의 값은 0.939693
sin(120)의 값은 0.866026
sin(130)의 값은 0.766045
sin(140)의 값은 0.642788
sin(150)의 값은 0.500000
sin(160)의 값은 0.342021
sin(170)의 값은 0.173649
sin(180)의 값은 0.000001
```

09. 난수(random number)는 컴퓨터를 이용한 문제 해결에서 많이 사용된다. 특히 수학적 분석이 너무 복잡한 경우에 시뮬레이션을 사용하면 실제로 제품을 제작하지 않고서도 많은 실험을 할 수 있다. Visual Studio의 경우, rand()가 한번 호출될 때마다 0에서 32767까지의 정수를 같은 확률로 선택하여 반환한다. rand() 함수를 이용하여 0 또는 1 값을 무작위로 반환하는 함수 b_rand()를 작성하고 5번 호출하여 보자.

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

int b_rand(void);

int main(void)
{
    int i;
    srand((unsigned)time(NULL));
    for (i = 1; i <= 5; i++)
    {
        printf("%d ", b_rand());
    }
    return 0;
}

int b_rand()
{
    return rand() % 2;
}
```

Microsoft Visual Studio 디버그 콘솔

1 1 0 1 0

10. 앞에서 작성한 b_rand() 함수를 이용하여 간단한 동전 던지기 게임을 시뮬레이션하여 보자. 컴퓨터가 동전을 던지고 사용자는 앞뒤를 말한다. 컴퓨터는 b_rand()를 이용하여 생성된 난수가 1이면 동전의 앞면으로 간주하고 0이면 동전의 뒷면으로 간주한다.

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

int b_rand(void);

int main(void)
{
    int i, num;
    char ch;
    srand((unsigned)time(NULL));

    do
    {
        i = b_rand();
        printf("앞면 또는 뒷면(1 또는 0):");
        scanf("%d", &num);
        if (num == i)
            printf("맞았습니다.\n");
        else
            printf("틀렸습니다.\n");

        printf("계속하시겠습니까?(y 또는 n):");

        while (1)
        {
            scanf("%c", &ch);
            if (ch == 'n' || ch == 'y')
                break;
        }
    } while (ch != 'n');
}

int b_rand(void)
{
    return rand() % 2;
}
```

Microsoft Visual Studio 디버그 콘솔

```
앞면 또는 뒷면(1 또는 0):1
맞았습니다.
계속하시겠습니까?(y 또는 n):y
앞면 또는 뒷면(1 또는 0):0
맞았습니다.
계속하시겠습니까?(y 또는 n):y
앞면 또는 뒷면(1 또는 0):1
틀렸습니다.
계속하시겠습니까?(y 또는 n):n
```

11. 0.0부터 1.0까지의 난수를 반환하는 함수 f_rand()를 작성하고 5번 호출하여 본다.

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

double f_rand(void);

int main(void)
{
    for (int i = 0; i < 5; i++)
        printf("%f ", f_rand());

    printf("\n");

    return 0;
}

double f_rand(void)
{
    return rand() / (double)RAND_MAX;
}
```

Microsoft Visual Studio 디버그 콘솔

0.001251 0.563585 0.193304 0.808741 0.585009

12. 화면에 세로로 막대 그래프를 그리는 프로그램을 작성하여 보자. 인수의 값만큼의 별표 문자를 출력하는 함수 `print_value(int n)`을 작성하라. 반복적으로 사용자로부터 값을 입력받아서 `print_value()`를 호출하여 입력값만큼 막대를 그리는 프로그램을 완성하라. 사용자가 음수를 입력하면 반복을 중단하라.

```
#include <stdio.h>

void print_value(int n);

int main(void)
{
    int n = 0;
    while (n >= 0)
    {
        printf("값을 입력하시오(종료는 음수): ");
        scanf("%d", &n);
        print_value(n);
    }
}

void print_value(int n)
{
    int i;
    for (i = 1; i <= n; i++)
    {
        printf("*");
    }
    printf("\n");
}
```

Microsoft Visual Studio 디버그 콘솔

값을 입력하시오(종료는 음수): 5

값을 입력하시오(종료는 음수): 10

값을 입력하시오(종료는 음수): -1

13. 두 개의 정수 n, m을 입력받아서 n이 m의 배수이면 1을 반환하고 그렇지 않으면 0을 반환하는 함수 is_multiple(int n, int m)를 작성하고 테스트하여 보자.

```
#include <stdio.h>

int is_multiple(int n, int m);

int main(void)
{
    int x, y;
    int result;
    printf("첫 번째 정수를 입력하시오:");
    scanf("%d", &x);
    printf("두 번째 정수를 입력하시오:");
    scanf("%d", &y);

    result = is_multiple(x, y);

    if (result == 1)
        printf("%d은 %d의 배수입니다.\n", x, y);
    else
        printf("%d은 %d의 배수가 아닙니다.\n", x, y);

    return 0;
}

int is_multiple(int n, int m)
{
    if (n % m == 0)
        return 1;
    else
        return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

```
첫 번째 정수를 입력하시오:30
두 번째 정수를 입력하시오:5
30은 5의 배수입니다.
```

14. 두 점 사이의 거리를 계산하는 함수를 작성하여 보자. 2차원 공간에서 두 점 사이의 거리를 계산하는 `get_distance(double x1, double y1, double x2, double y2)`를 작성하시오. 다음과 같은 두 점 사이의 거리를 계산하는 공식을 사용하라. 제공근은 `sqrt()` 라이브러리 함수를 사용하라.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

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

double get_distance(double x1, double y1, double x2, double y2);

int main(void)
{
    double x1, x2, y1, y2;
    printf("첫번째 점의 좌표를 입력하시오:(x, y)");
    scanf("%lf %lf", &x1, &x2);
    printf("두번째 점의 좌표를 입력하시오:(x, y)");
    scanf("%lf %lf", &y1, &y2);
    printf("두점 사이의 거리는 %lf입니다.\n", get_distance(x1, x2, y1, y2));
    return 0;
}

double get_distance(double x1, double y1, double x2, double y2)
{
    double d;
    d = sqrt(((x1 - x2) * (x1 - x2)) + ((y1 - y2) * (y1 - y2)));
    return d;
}
```

Microsoft Visual Studio 디버그 콘솔

```
첫번째 점의 좌표를 입력하시오:(x, y)1 1
두번째 점의 좌표를 입력하시오:(x, y)10 10
두점 사이의 거리는 12.727922입니다.
```

15. 주어진 정수가 소수인지를 검사하는 함수 `is_prime()`을 작성하라. 이 함수를 이용하여 2부터 100 사이의 모든 소수를 출력하라.

```
#include <stdio.h>

int is_prime(int x);

int main(void)
{
    int i;
    for (i = 2; i <= 100; i++)
    {
        is_prime(i);
    }
}

int is_prime(int x)
{
    int o;
    for (o = 2; o < x; o++)
    {
        if (x % o == 0)
            return 0;
    }
    printf("%d ", x);
    return 1;
}
```

Microsoft Visual Studio 디버그 콘솔

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

16. 오일러의 수 e 는 자연 로그의 밑수로 사용된다. 이 값은 다음과 같은 식에 의하여 근사치를 구할 수 있다. 본문에 있는 팩토리얼 값을 계산하는 함수 `factorial()`을 호출하여서 오일러의 수를 계산하는 프로그램을 작성하라.

$$e = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$$

```
#include <stdio.h>

void factorial(int x);

int main(void)
{
    int num;
    printf("어디까지 계산할까요: ");
    scanf("%d", &num);

    factorial(num);
}

void factorial(int x)
{
    int i, o, pt;
    double e = 0;

    for (i = 1; i <= x; i++)
    {
        pt = 1;
        for (o = 1; o <= i; o++)
        {
            pt *= o;
        }
        e += 1.0 / pt;
    }

    printf("오일러의 수는 %f입니다.\n", e + 1);
}
```

Microsoft Visual Studio 디버그 콘솔

어디까지 계산할까요: 30
오일러의 수는 2.718282입니다.

17. 두개의 부동 소수점 수가 근사적으로 같은 값이면 1을 반환하고 근사적으로 같지 않으면 0을 넘겨주는 함수 `f_equal(a, b)`를 작성하시오. 근사적으로 같다고 하는 것은 다음의 부등식을 만족하는 경우라고 가정한다.

$$|a-b|/\min(|a|, |b|) < e$$

```
#include <stdio.h>
#define e 0.000001

int f_equal(double a, double b);
double f_min(double x, double y);
double f_abs(double x);

int main(void)
{
    double x, y;
    printf("실수를 입력하시오: ");
    scanf("%lf", &x);
    printf("실수를 입력하시오: ");
    scanf("%lf", &y);

    if (f_equal(x, y) == 1)
        printf("두 개의 실수는 서로 같음\n");
    else
        printf("두 개의 실수는 서로 다름\n");
}

int f_equal(double a, double b)
{
    if (f_abs(a) - f_abs(b) / f_min(f_abs(a), f_abs(b)) < e)
        return 1;
    return 0;
}

double f_min(double x, double y)
{
    return x > y ? y : x;
}

double f_abs(double x)
{
    return x;
}
```

Microsoft Visual Studio 디버그 콘솔

```
실수를 입력하시오: 1.2345
실수를 입력하시오: 1.2346
두 개의 실수는 서로 다름
```

18. 사용자로부터 2개의 숫자를 받아서 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지 연산의 결과를 계산해주는 프로그램을 작성해보자. 다음과 같은 메뉴를 화면에 표시한다. 프로그램 작성 시에 최대한 함수를 많이 사용해보자.

```
#include <stdio.h>

int add(int a, int b);
int minus(int a, int b);
int multiply(int a, int b);
int divide(int a, int b);
int rest(int a, int b);

int main(void)
{
    int x = 0, a, b, result;
    char ch;

    printf("=====MENU=====\n");
    printf("1. 덧셈\n2. 뺄셈\n3. 곱셈\n4. 나눗셈\n5. 나머지\n");

    do
    {
        printf("원하는 메뉴를 선택하시오(1-5):");
        scanf("%d", &x);

        printf("숫자 2개를 입력하시오: ");
        scanf("%d %d", &a, &b);

        if (x == 1)
        {
            result = add(a, b);
            printf("연산결과 : %d\n", result);
        }
        else if (x == 2)
        {
            result = minus(a, b);
            printf("연산결과 : %d\n", result);
        }
        else if (x == 3)
        {
            result = multiply(a, b);
            printf("연산결과 : %d\n", result);
        }
        else if (x == 4)
        {
            result = divide(a, b);
            printf("연산결과 : %d\n", result);
        }
        else
        {
            result = rest(a, b);
            printf("연산결과 : %d\n", result);
        }
    } while (ch != 'q');
```



```

    {
        result = rest(a, b);
        printf("연산결과 : %d\n", result);
    }

    printf("계속하려면 y를 누르시오:");

    while (1)
    {
        scanf("%c", &ch);
        if (ch == 'y')
            break;
    }
} while (ch == 'y');
}

```

```

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

```

```

int minus(int a, int b) {
    int result = a - b;
    return result;
}

```

```

int multiply(int a, int b) {
    int result = a * b;
    return result;
}

```

```

int divide(int a, int b) {
    int result = a / b;
    return result;
}

```

```

int rest(int a, int b) {
    int result = a % b;
    return result;
}

```

```

=====
MENU
=====

```

```

1. 덧셈
2. 뺄셈
3. 곱셈
4. 나눗셈
5. 나머지
원하는 메뉴를 선택하시오(1-5): 1
숫자 2개를 입력하시오: 10 20
연산결과 : 30
계속하려면 y를 누르시오:

```