

REPORT DESIGN

REPORT DESIGN

REPORT

C프로그래밍 기말고사 대체 과제 7장~10장

학 과 컴퓨터정보공학과
학 번 20200814
이 름 임유빈
반 A반

CHAPTER 7

반복

02 다음을 출력하는 프로그램을 for 문을 이용하여 작성하시오.

```
1
21
321
4321
54321
```

소스코드

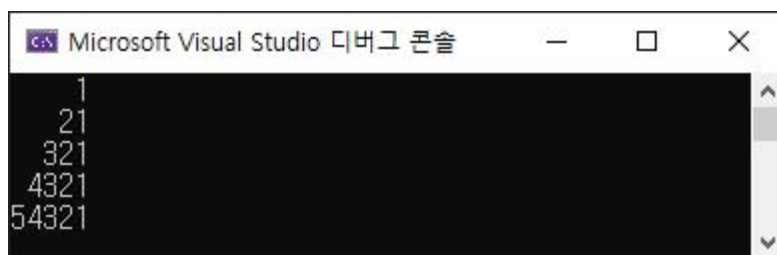
```
//학번: 20200814, 이름: 임유빈

#include <stdio.h> //stdio.h 헤더파일 추가

int main(void)
{
    for (int a = 1; a <= 5; a++) //한 줄을 출력하는 반복문
    {
        for (int b = 5; b >= 1; b--) // 처음 부분부터 5번째 자리까지 출력해주는 반복문
        {
            if (a < b)
                printf(" "); //만약 a가 1이면 b가 2까지는 크기 때문에 스페이스 4번 띄워줌
            else
                printf("%d", b); //만약 a가 1이면 b가 1일 때 같으므로 1출력 "    1"
        }

        puts(""); //한 줄 내림
    }
}
```

결과



```
Microsoft Visual Studio 디버깅 콘솔
1
21
321
4321
54321
```

06 다음을 출력하는 프로그램을 중첩된 for 문을 이용하여 작성하시오.

```

0
101
21012
3210123
432101234
54321012345
6543210123456
765432101234567
    
```

소스코드

//학번: 20200814, 이름: 임유빈

#include <stdio.h> //stdio.h 헤더파일 추가

```

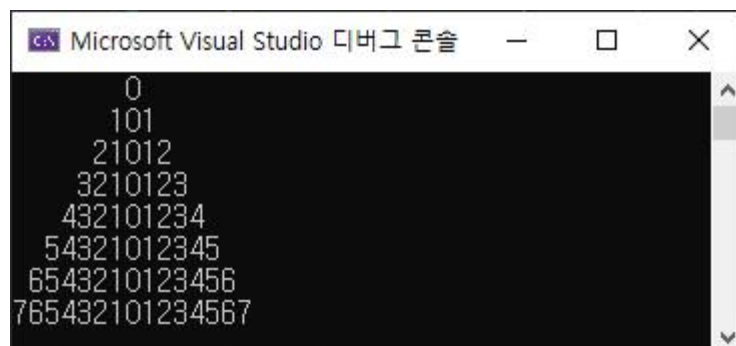
int main(void)
{
    for (int a = 0; a <= 7; a++) //한 줄 출력하는 반복문
    {
        for (int b = 7; b >= 0; b--) // 처음 부분부터 8번째 자리까지 출력해주는 반복문
        {
            if (a < b)
                printf(" "); //만약 a가 1이면 b가 2까지는 크기 때문에 스페이스 6번 띄워줌
            else
                printf("%d", b); //만약 a가 1이면 b가 1과 0일 때 같거나 작으므로 1과 0 출력 "      10"

        }

        for (int c = 1; c <= a; c++) // 9번째 자리부터 출력해주는 반복문
            printf("%d", c); //만약 a가 1이면 c는 a보다 작거나 같은 숫자가 1밖에 없으므로 1을 출력 "      10'1"

        puts(""); //한 줄 내림
    }
}
    
```

결과



11 표준입력으로 받은 9이하의 정수로 구구단을 출력하는 프로그램을 작성하시오.

소스코드

```
//학번: 20200814, 이름: 임유빈

#define _CRT_SECURE_NO_WARNINGS //scanf() 오류 방지하기 위한 상수 정의

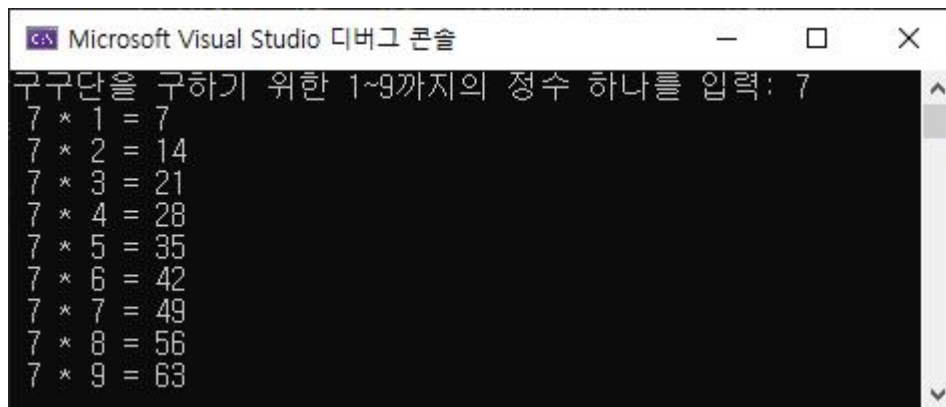
#include <stdio.h> //stdio.h 헤더파일 추가

int main(void)
{
    int num; //자료형이 int인 num 변수 선언

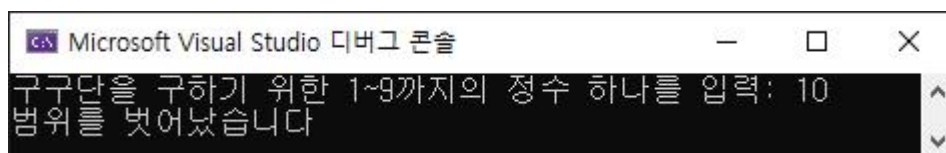
    printf("구구단을 구하기 위한 1~9까지의 정수 하나를 입력: "); //입력할 정보 표시
    scanf("%d", &num); //num에 입력받음

    if (num >= 1 && num <= 9) //num의 값이 9 이하에 양의 정수인가 판별하는 조건문
    {
        for (int i = 1; i <= 9; i++) //곱해주는 값을 반복문으로 처리
        {
            printf(" %d * %d = %d\n", num, i, num * i);
        }
    }
    else
        printf("범위를 벗어났습니다\n"); //9이하의 양의정수 범위를 벗어나는 경우 출력
}
```

결과



```
Microsoft Visual Studio 디버그 콘솔
구구단을 구하기 위한 1~9까지의 정수 하나를 입력: 7
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
```



```
Microsoft Visual Studio 디버그 콘솔
구구단을 구하기 위한 1~9까지의 정수 하나를 입력: 10
범위를 벗어났습니다
```

CHAPTER 8

포인터 기초

02 다음을 참고로 int 형 포인터에 관한 프로그램을 작성하시오.

- 두 정수를 저장하기 위해 변수를 선언하여 각각 10,20을 저장

```
int data1 = 10, data2 = 20;
```

- 다음 변수 sum과 포인터 변수 p를 선언한 후 p와 sum 만을 사용하여 data1에는 100을 data2에는 200을 저장한 후 위의 두 변수의 합을 변수 sum에 저장하여 출력

소스코드

//학번: 20200814, 이름: 임유빈

#include <stdio.h> //stdio.h 헤더파일 추가

int main(void)

{

int data1 = 10, data2 = 20; //data1 변수에 10을 data2 변수에 20을 저장

int sum = 0; //합을 구하는 변수 0으로 초기화

int* p; //포인터 p 선언

p = &data1; //p는 data1의 주소를 가리킨다

*p = 100; // *p는 data1의 주소가 아닌 저장된 값을 의미 data1 = 100; 동일

sum += *p; //sum에다가 data1의 값 즉 100을 더하여 저장 ---- 100

p = &data2; //p는 data2의 주소를 가리킨다

*p = 200; // *p는 data2의 주소가 아닌 저장된 값을 의미 data2 = 200; 동일

sum += *p; //sum에다가 data2의 값 즉 200을 더하여 저장 ---- 300

printf("data1 + data2 = %d\n", sum); //합계 출력

return 0;

}

결과

```
Microsoft Visual Studio 디버그 콘솔
data1 + data2 = 300
```

06 다음과 같이 int 형 변수 i에 0x3C405B7B를 정의하고, char* 변수 p를 선언한 후, 포인터 변수 p가 다음 내용 5B인 문자 '['를 가리키도록 하여 다음 조건이 만족하는 프로그램을 작성하시오.

- int n = 0x3C405B7B;
- 변수 n에 저장된 각 바이트를 분석하면 다음과 같으며 포인터 p가 주소 p를 가리키도록

주소 p+2	주소 p+1	주소 p	주소 p-1
3C (<)	40 (@)	5B ([)	7B ({)

소스코드

//학번: 20200814, 이름: 임유빈

#include <stdio.h> //stdio.h 헤더파일 추가

int main(void)

{

int n = 0x3C405B7B; //n 변수에 0x3C405B7B 저장

char* p = (char*)&n + 1; //포인터 p가 n의 주솟값에다가 +1한 주소를 가리키도록

//여기서 n의 자료형이 int지만 자료형이 다른 포인터를 정의하고 싶다면 포인터 형변환을 통해 n의 주소를 가리키는 것이 가능

printf("p의 주소가 가리키는 값: %c\n", *p); // *p의 주솟값이 아닌 내용물

printf("p-1의 주소가 가리키는 값: %c\n", *(p - 1)); // *(p - 1)의 주솟값이 아닌 내용물

printf("p+1의 주소가 가리키는 값: %c\n", *(p + 1)); // *(p + 1)의 주솟값이 아닌 내용물

printf("p+2의 주소가 가리키는 값: %c\n", *(p + 2)); // *(p + 2)의 주솟값이 아닌 내용물

//각각의 주소는 p, (p - 1), (p + 1), (p + 2)

//printf("*p++의 주소가 가리키는 값: %c\n", *p++); // [

//printf("*(p++)의 주소가 가리키는 값: %c\n", *(p++)); // [

//printf("+++p의 주소가 가리키는 값: %c\n", +++p); // @

//printf("*(++p)의 주소가 가리키는 값: %c\n", *(++p)); // @

//printf("(p)++의 주소가 가리키는 값: %c\n", (p)++); // [

//printf("++*p의 주소가 가리키는 값: %c\n", ++*p); // \

//printf("++(*p)의 주소가 가리키는 값: %c\n", ++(*p)); // \

return 0;

}

결과

```

Microsoft Visual Studio 디버그 콘솔
p의 주소가 가리키는 값: [
p-1의 주소가 가리키는 값: {
p+1의 주소가 가리키는 값: @
p+2의 주소가 가리키는 값: <
  
```


- 다음 표에서 결과값을 위의 값에서 각각 출력하도록 하여 빈 부분을 채우시오.
※이 문제의 답은 빨간색으로 표시했습니다.

연산식		결과값 의미	위 자료에서 실제 결과값
*p++	*(p++)	*p: p의 간접참조 값	[
+++p	*(++p)	*(p+1): (p+1) 간접참조 값	@
(*p)++		*p: p의 간접참조 값	[
++*p	++(*p)	*p + 1: p의 간접참조 값에 1 증가	\

CHAPTER 9

배열

02 다음을 참고로 배열에서 모든 원소의 값을 모두 10씩 증가시키는 프로그램을 작성하시오.

- 배열 `int data[] = {3,21,35,57,24,82,8};`
- 모든 원소에서 10씩 증가시키기 위해서 `*(data + i)`와 반복문을 사용

소스코드

//학번: 20200814, 이름: 임유빈

#include <stdio.h> //stdio.h 헤더파일 추가

int main(void)

{

int data[] = { 3,21,35,57,24,82,8 }; //data[] 배열 안에 7개의 원소 정의

int sizeonedata = sizeof(data) / sizeof(data[0]); //data[]의 원소의 개수

for (int i = 0; i < sizeonedata; i++) //반복문으로 data[] 안에 있는 원소값 출력

printf("%d ", data[i]);

puts(""); //한 줄 내림

printf("배열에서 모든 원소값을 10을 더해준 후: \n");

for (int i = 0; i < sizeonedata; i++) //반복문으로 원소값들을 각각 10씩 더해줌

*(data + i) += 10; //data[i] += 10; 동일

for (int i = 0; i < sizeonedata; i++) //반복문으로 data[] 안에 있는 원소값 출력

printf("%d ", data[i]);

puts(""); //한 줄 내림

return 0;

}

결과

```

Microsoft Visual Studio 디버그 콘솔
3 21 35 57 24 82 8
배열에서 모든 원소값을 10을 더해준 후:
13 31 45 67 34 92 18
    
```

06 다음과 같이 일차원 배열의 동등함을 검사하여 그 결과를 알아보는 프로그램을 작성하시오.

- 다음 배열 a와 b에 대하여 검사

```
int a[] = { 4, 7, 9, 3, 6 };
int b[] = { 4, 7, 9, 3, 6 };
int b[] = {10, 20, 30, 40, 50};
int b[] = { 4, 7, 9, 3, 7 };
```

- 배열 a와 b의 배열크기가 다르면 다른 배열이며, 같으면 순차적으로 원소값이 모두 같으면 ‘같은 배열’이고, 하나라도 다르면 ‘다른 배열’

소스코드 ※깁니다.

//학번: 20200814, 이름: 임유빈

#include <stdio.h> //stdio.h 헤더파일 추가

//배열의 비교에서 크기나 순서, 값들이 같은지 판별하고 출력해주는 함수원형

void printarray(int); //void printarray(int n) 가능

int main(void)

{

int equals; //배열을 비교하는 데에 따라서 변화되는 변수 printarray() 함수와 함께 사용됨

int a[] = { 4,7,9,3,6 }; //a[] 배열 안에 5개의 원소 정의

int b1[] = { 4,7,9,3,6 }; //b1[] 배열 안에 5개의 원소 정의

int b2[] = { 10,20,30,40,50 }; //b2[] 배열 안에 5개의 원소 정의

int b3[] = { 4,7,9,3,7 }; //b3[] 배열 안에 5개의 원소 정의

int sizeoneA = sizeof(a) / sizeof(a[0]); //a[]의 전체 크기는 int형이기 때문에 20바이트 원소 하나의 크기는 4바이트
//배열의 총 크기에서 원소 하나의 포인터 크기를 나누면 원소의 개수가 된다. 즉 여기서는 20/4 == 5가 된다.

int sizeoneB1 = sizeof(b1) / sizeof(b1[0]); //b1[]의 원소의 개수

int sizeoneB2 = sizeof(b2) / sizeof(b2[0]); //b2[]의 원소의 개수

int sizeoneB3 = sizeof(b3) / sizeof(b3[0]); //b3[]의 원소의 개수

printf("int a[] = { %d,%d,%d,%d,%d }\n", a[0], a[1], a[2], a[3], a[4]); //a[]의 정보를 출력

printf("int b1[] = { %d,%d,%d,%d,%d }\n", b1[0], b1[1], b1[2], b1[3], b1[4]); //b1[]의 정보를 출력

printf("int b2[] = { %d,%d,%d,%d,%d }\n", b2[0], b2[1], b2[2], b2[3], b2[4]); //b2[]의 정보를 출력

printf("int b3[] = { %d,%d,%d,%d,%d }\n", b3[0], b3[1], b3[2], b3[3], b3[4]); //b3[]의 정보를 출력

puts("-----");

//배열을 비교할 때 같은가를 판별하는 조건문

if (sizeof(a) == sizeof(b1)) //a[]와 b1[]의 전체 크기가 같은가를 묻는 조건

{

if (sizeoneA == sizeoneB1) //a[]와 b1[] 안에 있는 원소의 개수가 같은가를 묻는 조건

{

for (int i = 0; i < sizeoneA; i++) //sizeoneA가 5이므로 0~4까지 총 5번 실행하는 반복문

{

if (a[i] == b1[i]) //a[] 안에 있는 원소값과 b1[] 안에 있는 원소값이 같은가를 묻는 조건

```

        equals = 1;
    }
    else
    {
        equals = 0; //equals라는 변수에 0을 저장
        break; //이때 가장 안쪽에 있는 조건문이 한 번이라도 거짓이 나오면
    } //break: 때문에 반복문을 바로 빠져나옴
}
}
else
    equals = 0;
}
else
    equals = 0;

```

```

printf("a[]와 b1[]은 서로 ");
printarray(equals); //equals이 1인 경우 "같은 배열", 0인 경우 "다른 배열"

```

//위와 같다

```

if (sizeof(a) == sizeof(b2)) //a[]와 b2[]의 전체 크기가 같은가를 묻는 조건
{
    if (sizeoneA == sizeoneB2) //a[]와 b2[] 안에 있는 원소의 개수가 같은가를 묻는 조건
    {
        for (int i = 0; i < sizeoneA; i++)
        {
            if (a[i] == b2[i])
                equals = 1;
            else
            {
                equals = 0;
                break;
            }
        }
    }
    else
        equals = 0;
}
else
    equals = 0;

```

```

printf("a[]와 b2[]은 서로 ");
printarray(equals);

```

//위와 같다

```

if (sizeof(a) == sizeof(b3)) //a[]와 b3[]의 전체 크기가 같은가를 묻는 조건
{
    if (sizeoneA == sizeoneB3) //a[]와 b3[] 안에 있는 원소의 개수가 같은가를 묻는 조건
    {
        for (int i = 0; i < sizeoneA; i++)
        {
            if (a[i] == b3[i])
                equals = 1;
            else
            {
                equals = 0;
                break;
            }
        }
    }
    else
        equals = 0;
}
else
    equals = 0;

```

```

    }
}
else
    equals = 0;
}
else
    equals = 0;

printf("a[]와 b3[]은 서로 ");
printarray(equals);

return 0;
}

```

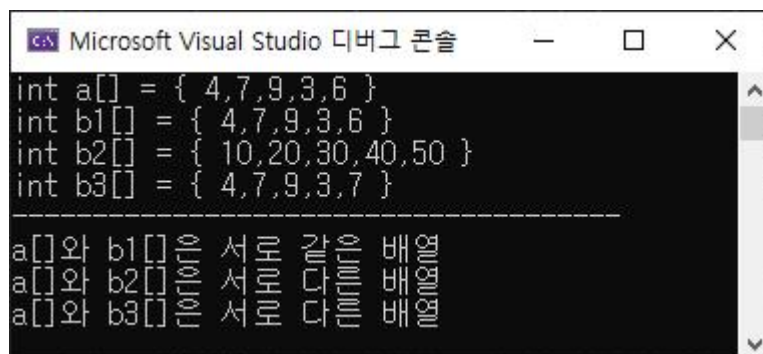
//배열의 비교에서 크기나 순서, 값들이 같은지 판별하고 출력해주는 함수 구현

```

void printarray(int n)
{
    if (n == 1)
        printf("같은 배열\n"); //equals == 1이면 "같은 배열"을 출력, equals == 0이면 "다른 배열"을 출력
    else
        printf("다른 배열\n");
}

```

결과



```

Microsoft Visual Studio 디버그 콘솔

int a[] = { 4,7,9,3,6 }
int b1[] = { 4,7,9,3,6 }
int b2[] = { 10,20,30,40,50 }
int b3[] = { 4,7,9,3,7 }

-----
a[]와 b1[]은 서로 같은 배열
a[]와 b2[]은 서로 다른 배열
a[]와 b3[]은 서로 다른 배열

```

CHAPTER 10

함수 기초

02 두 개의 임의 정수 m,n을 입력 받아 다음 함수를 작성하여 mn의 결과를 출력하는 프로그램을 작성하시오.

- 함수 `intpow(int m, int n)`의 결과는 `mn`

소스코드

```
//학번: 20200814, 이름: 임유빈

#define _CRT_SECURE_NO_WARNINGS //scanf() 오류 방지하기 위한 상수 정의

#include <stdio.h> //stdio.h 헤더파일 추가

//두 정수를 붙여서 보여주는 함수원형
void intpow(int m, int n); //void intpow(int, int) 도 가능 //매개변수 두 개 //반환형은 없음

int main(void)
{
    int m, n; //임의 정수 m, n 변수 선언

    printf("두 정수를 입력하시오 : "); //입력할 두 정수 정보
    scanf("%d %d", &m, &n); //m과 n에 입력받음
    puts(""); //한 줄 내림

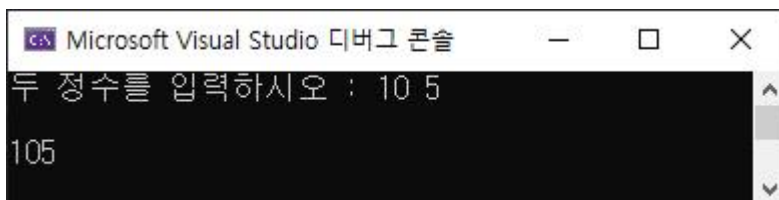
    intpow(m, n); //함수 intpow() 호출

    return 0;
}

//두 정수를 붙여서 보여주는 함수 구현
void intpow(int m, int n)
{
    printf("%d%d\n", m, n); //mn을 출력

    return; //반환형이 void이므로 아무것도 반환하지 않음 //안 써줘도 가능
}
```

결과



06 다음을 참고로 실수 r 을 입력 받아 이 값을 반지름으로 하는 원의 면적과 둘레의 길이를 구하는 프로그램을 작성하시오.

- 면적을 구하는 함수의 이름은 `area()`로, 원의 둘레를 구하는 함수의 이름은 `circumference()`로 하고, 원주율을 3.14로 매크로로 정의
- 원의 면적을 구하는 식은 원주율(3.14) $\times r \times r$ 이며 원의 둘레는 $2 \times \text{원주율}(3.14) \times r$

소스코드

```
//학번: 20200814, 이름: 임유빈

#define _CRT_SECURE_NO_WARNINGS //scanf() 오류 방지하기 위한 상수 정의

#include <stdio.h> //stdio.h 헤더파일 추가
#define PI 3.14 //원주율 3.14를 매크로 상수 PI로 정의

//원의 면적을 구하는 함수원형
double area(double x); //double area(double); 도 가능 //매개변수 하나 //반환형은 double
//원의 둘레의 길이를 구하는 함수원형
double circumference(double x); //double circumference(double); 도 가능 //매개변수 하나 //반환형은 double

int main(void)
{
    double r; //반지름 r 변수 실수형으로 선언

    printf("원의 반지름(실수) 입력: "); // 입력할 정보
    scanf("%lf", &r); // 실수 r을 입력받음
    puts(""); //한 줄 내림

    printf("원의 면적 : %.2lf\n", area(r)); //area()의 호출로 원의 면적을 저장하고 출력
    printf("원의 둘레의 길이 : %.2lf\n", circumference(r)); //circumference()의 호출로 원의 둘레의 길이를 저장하고 출력

    return 0;
}

//원의 면적을 구하는 함수 구현
double area(double x)
{
    double CircleArea = PI * x * x; //CircleArea라는 변수에 원의 면적을 저장

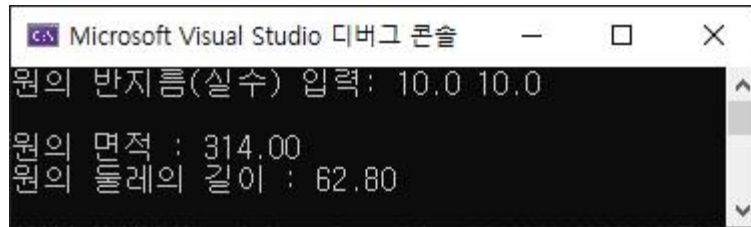
    return CircleArea; //원의 면적을 리턴
}

//원의 둘레의 길이를 구하는 함수 구현
double circumference(double x)
{
    double CircleCircumference = 2 * PI * x; //CircleCircumference라는 변수에 원의 둘레의 길이를 저장
```

```
return CircleCircumference; //원의 둘레의 길이를 리턴
```

```
}
```

결과



11 x의 y승을 구하는 함수를 재귀적으로 만들고, 6의 0승에서 10승까지 출력하는 프로그램을 작성하시오.

소스코드

```
//학번: 20200814, 이름: 임유빈

#include <stdio.h> //stdio.h 헤더파일 추가

//제곱을 재귀적으로 처리해주는 함수원형
int repeatPow(int, int); //int repeatPow(int num, int pNum) 가능

int main(void)
{
    int num = 6; //num 변수의 6으로 초기화

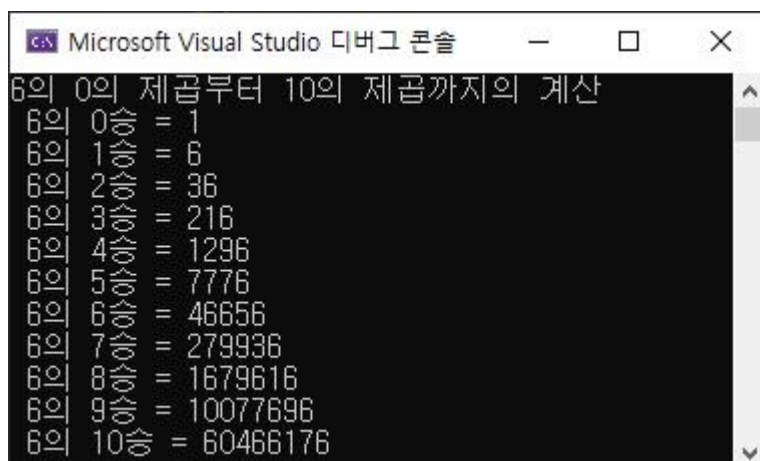
    printf("6의 0의 제곱부터 10의 제곱까지의 계산\n");

    for (int i = 0; i <= 10; i++) //0승에서 10승까지 출력해주는 반복문
    {
        printf(" %d의 %d승 = %d\n", num, i, repeatPow(num, i));
        //repeatPow(num 인자값 6을 받음, i 인자값 반복문 0~10까지 받음)
    }
}

//제곱을 재귀적으로 처리해주는 함수 구현
int repeatPow(int num, int i)
{
    if (i == 0)    //만약 반복문에서 i의 값이 0이면 항상 1로 리턴 숫자의 0승은 항상 1이기 때문에
        return 1;

    else
        return num * repeatPow(num, i - 1); //repeatPow() 함수를 repeatPow() 함수 안에서 또 호출(재귀적)
}
```

결과



```
Microsoft Visual Studio 디버그 콘솔
6의 0의 제곱부터 10의 제곱까지의 계산
6의 0승 = 1
6의 1승 = 6
6의 2승 = 36
6의 3승 = 216
6의 4승 = 1296
6의 5승 = 7776
6의 6승 = 46656
6의 7승 = 279936
6의 8승 = 1679616
6의 9승 = 10077696
6의 10승 = 60466176
```