C프로그래밍(HW04)

2011920026

컴퓨터과학부

서양훈

P284

문제 12-1

문제 1

/\*

num의 주소 : 0x12 라고 가정.

num의 값 : 10

ptr1 : 0x12 (ptr1 = &num)

ptr2 : 0x12 (ptr2 = ptr1; 으로 인해 ptr1에 저장된 주소값, 즉 num의 주소값이 ptr2에 저장됨.)

ptr1 => num <= ptr2 (ptr1과 ptr2가 num을 가리킨다.(==ptr1과 ptr2에 num의 주소값이 저장되어있다.)

ptr1가 가리키는 int형 변수(num): num의 값 1만큼 후위증가

ptr2가 가리키는 int형 변수(num): num의 값 1만큼 후위증가

num의값은 2가 증가되어 (num = 12)

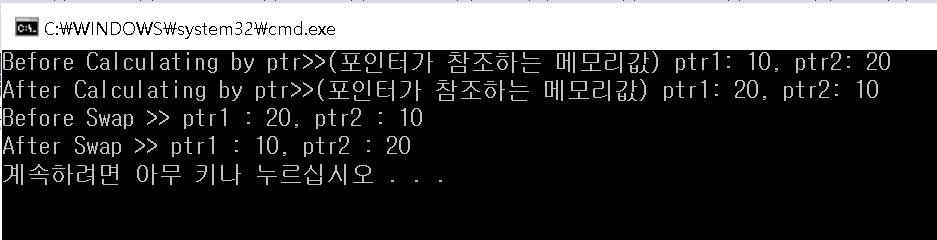
12 출력.

\*/

문제 2

시행목적 : 포인터를 이용해 계산전과 계산후의 결과를 비교

시행결과



토론 : 포인터 참조연산시 \*연산을 통해 주소가 가리키는 메모리에 직접접근하여 값을 참조할 수 있다.

P299

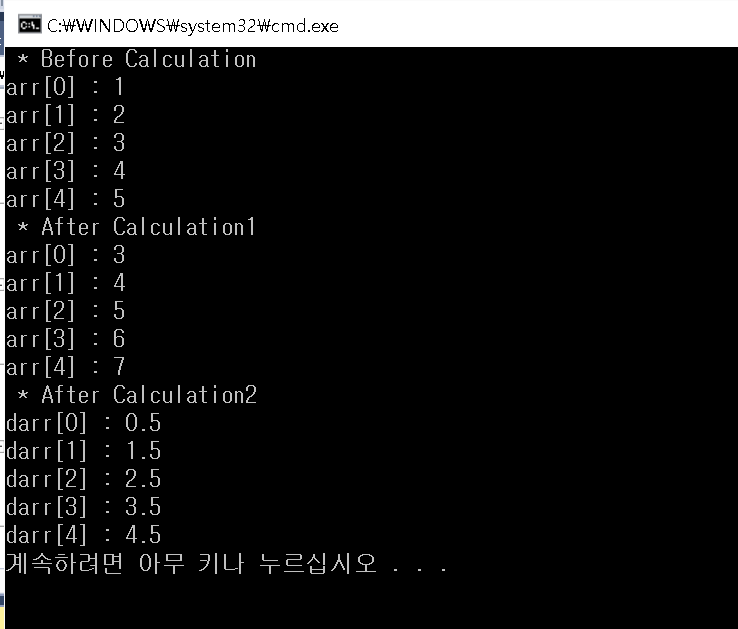
문제 13-1

문제 1

시행목적 : 계산전과 계산후의 결과를 비교하고, 정수형배열이 아닌 실수형배열을 이용해 실수의 값도 출력. + ptr을 이용해 포인터증감연산으로 메모리에 직접 접근하는지 평가

(ptr저장된 값 변경함.)

시행결과



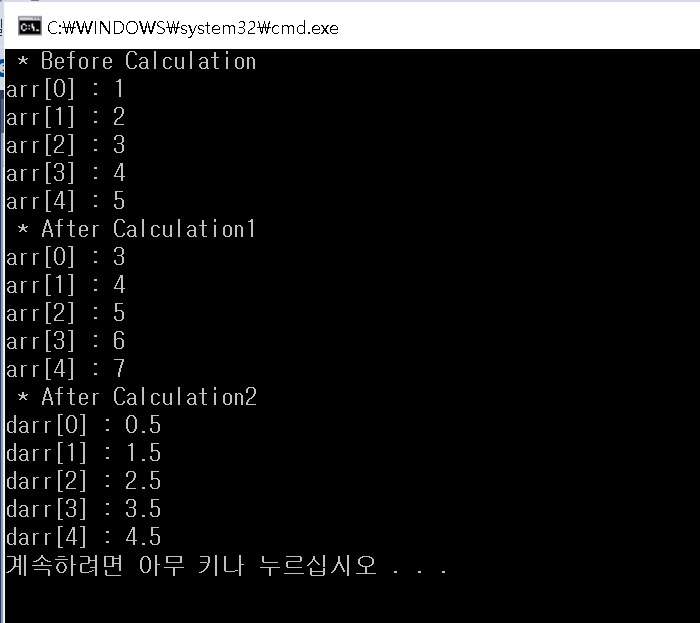
토론 : sizeof() 연산자를 이용해 배열의 길이를 구할 수 있다.(len= sizeof(arr) / sizeof(int))

문제 2

시행목적 : 계산전과 계산후의 결과를 비교하고, 정수형배열이 아닌 실수형배열을 이용해 실수의 값도 출력. + ptr을 이용해 포인터증감연산으로 메모리에 직접 접근하는지 평가

(ptr저장된 값 변경 안 함.)

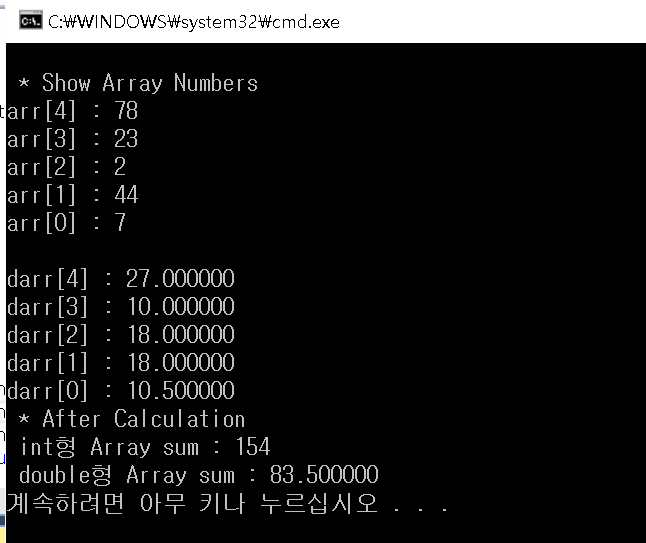
시행결과



토론 : for문 내에서 \*(ptr+i) += 2; 이런 식으로 작성하면, ptr저장된값을 변경하지 않고, 배열의 값을 변경시킬수 있다.

문제 3

시행목적 : int형말고도, double형도 추가

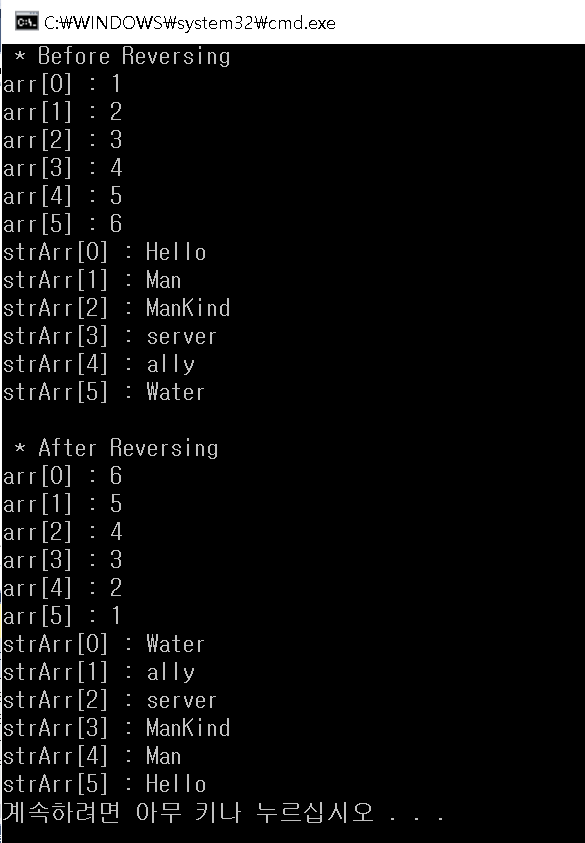
시행결과 

토론 : 포인터변수 parr에서 배열의 마지막 주소값을 받으므로, \*(parr--) = 값 의 형태로 for문 안에 작성한다.

문제 4

시행목적 : int형 정수 말고도, 문자열도 적용

시행결과



토론 : void \*\* strArr을 이용해

char\*형 포인터를 원소로 가지는 배열포인터(char\*\*)를 인자로 받고,

인자로 받은 함수 내에서 (char \*\* strArr, char \* front, char \* rear)을 이용해

저장된 문자열 순서를 뒤집을 수 있다.

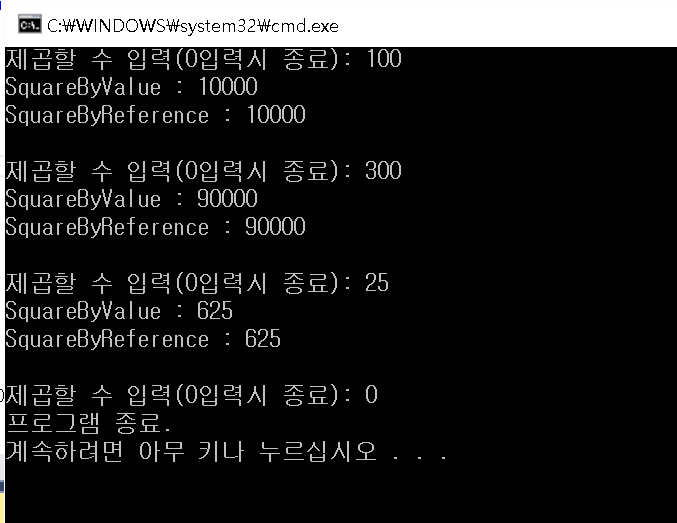
P320

문제 14-1

문제 1

시행목적 : 한번입력하고 프로그램이 종료되지 않도록, 반복문을 이용해 계속 원하는 값을 입력하면 제곱한값을 출력.

시행결과

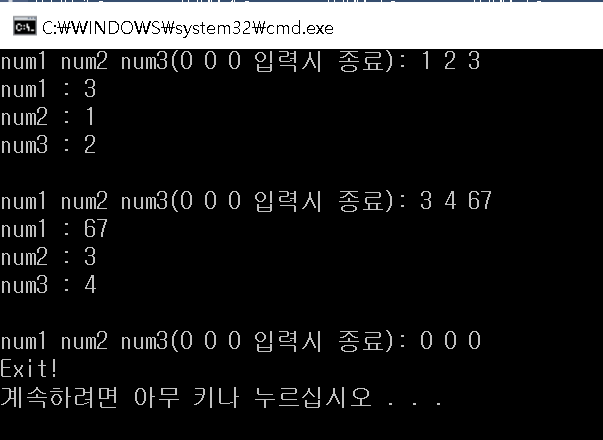


토론 : int형범위내에서 구현했고, 만약 제곱한 결과값이 int형 범위를 넘을 경우,

overflow!를 출력하고 0을 반환하게끔 해서 overflow여부를 알 수 있게끔 구현.

문제 2

시행목적 : 반복문을 이용해 새로운값을 입력하고 바꿀수있게끔 하기 위함.(일종의 메뉴)

시행결과 

토론 : tmp변수를 3개 만들어서 3개의값(num1, num2, num3)을 순서대로 변경할수 있다.

P324

문제 14-2

문제 1

/\*

문제 1

매개변수 arr포인터를 대상으로

const 선언을 한 이유는

showAllData의 함수 내에서는

Data의 변경없이,

Data를 출력만 해 주기 위해

값의 변경을 막도록

const선언을 해 주었다.

\*/

문제 2

/\*

문제 2

ptr이 참조하는 메모리값을 변경하지 않도록

ptr을 const선언해주었는데( const int \* ptr )

ShowData함수 정의부분에서

포인터rptr에 포인터ptr의 값을 복사해서,

ptr이 참조하는 메모리값을 변경하려고 하는

\*rptr = 20; 로 인해,

const선언해준것이 무의미하게 되므로 잘못되었고

컴파일러는 경고메세지를 띄운다.

\*/