프로그램 분석 문제 1

과제코드 + 주석

#include <stdio.h>

int reset(void); *// reset() 함수 원형 선언*

int next(int); *// next() 함수 원형 선언*

int index = 1; *// 변수의 수형, 이름, 초기값은 수정 불가*

void main(void) {

*/\**

*for문 안에 있는 반복문에서 출력을 지역변수 index으로 할 경우*

*1*

*1*

*1*

*가 출력이 될 것 같습니다.*

*\*/*

auto int index, jdex; *// 변수의 수형, 이름, 초기값은 수정 불가*

index = reset(); *//reset()함수를 통해 전역변수에 있는 index로 main함수 내에 있는 index를 초기화해줍니다*

for(jdx = 1; jdex <= 3; jdex++)*//jdx는 next()의 지역변수이기 때문에 main()에서 사용할 수 없습니다 => jdex로 수정*

{

printf('%d \n', indx, jdx, next(index));

*///printf에서 %d 하나만 써주었기 때문에 출력할 변수의 수를 줄이든가 %d를 추가해 나머지 변수들을 출력할 수 있게합니다.*

*// printf를 할 때는 작은따옴표가 아닌 큰따옴표로 감싸줘야 합니다.*

*//변수의 이름이 indx가 아니라 index로 해주어야 합니다.*

*//변수의 이름이 jdx 아니라 jdex 해주어야 합니다.*

}

}

int reset(void)

{*//이 함수는 전역변수 index를 반환합니다.*

return (index); *// main 안에 있는 지역변수 index가 아닌 함수 원형 밑에 선언 되어 있는 전역번수를 의미합니다*

}

int next(int jdx)

{*//next()의 파라미터 jdx에 전역변수 index를 1 더한 후 그 값으로 jdx를 초기화해줍니다*

jdx = ++index;

*//return으로 int값을 반환을 해주어야 하는데 해주지를 않았습니다.*

}*//함수의 반환형이 int인데 반환해주는 것이 없기 때문에 오류가 발생합니다.*

수정 코드 및 주석

#include <stdio.h>

int reset(void); *// reset() 함수 원형 선언*

int next(int); *// next() 함수 원형 선언*

int index = 1; *// 변수의 수형, 이름, 초기값은 수정 불가*

void main(void) {

int index, jdex; *// 변수의 수형, 이름, 초기값은 수정 불가*

index = reset();

for(jdex = 1; jdex <= 3; jdex++)

{

printf("%d\n", index); *//printf를 통해 지역변수 index를 출력*

next(index);

*//지역 변수의 값을 바꾸고 싶으면 파라미터를 넘길 때*

*// 그냥 변수명만 넘기는 것이 아닌 주소값으로 넘겨야 다른 함수에서 수정이 일어나면*

*// 그 주소의 값이 수정되기 때문에 지역변수의 값도 수정되어 있습니다.*

*//하지만 이번 과제의 코드의 경우 함수원형을 수정할 수 없으므로 next()함수로*

*//지역변수 index의 값은 바꿀 수 없습니다.*

*//값을 더한 수를 출력하고 싶으면 index = next(index); 로 지역변수값을 초기화 시키거나*

*//printf("%d\n", next(index)); 이렇게 출력해야 합니다.*

}

return 0;

}

int reset(void){

return (index);

}

int next(int jdx){

jdx = ++index;

return jdx;

}

출력 결과

main() 안에 있는 index가 next()함수로 인해 값이 증가되는 것 같지만

실제로는 값의 변화가 없어 결국 처음 초기화된 1로 반복분의 횟수인 3번 줄을 바꿔가며 출력됩니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

프로그램 분석 문제 2

과제코드 + 주석

#include <stdio.h>

int arr[] = { 1, 2, 3, 4, 5 }; *// 명칭, 초기값 변경 불가*

void main(void)

{

int inx, \*ptr; *// 변수의 명 수정 불가. // inx는 그냥 4바이드의 정수형 변수이고 ptr의 경우 그 변수의 메모리 공간주소를 가리키는 변수입니다*

for(in = 1; inx < 5; inx++) *//in을 inx로 수정합니다*

printf(‘%d, %d \n’, arr[inx]);

*// 작은 따옴표가 아닌 큰 따옴표로 묶어줍니다*

*// 따옴표 안에 %d하나를 지워줍니다.*

*/\* 출력 예상*

*2*

*3*

*4*

*5*

*\*/*

for(ptr = &arr[1]; ptr <= &arr[4 ; ptr++)

*//범위를 지정하는 곳에 arr의 변수의 대괄호를 닫아줍니다.*

*//이 반복문의 범위는 arr[1]인덱스의 주소부터 arr[4]인덱스 주소까지 입니다.*

printf(‘%d, %d \n’, \*ptr);

*//포인터배열의 주소안에 있는 값을 가져올 때는 변수이름 앞에 에스터리스크(\*)를 붙여줍니다.*

*/\*출력 예상*

*2*

*3*

*4*

*5*

*\*/*

}

수정 코드

#include <stdio.h>

int arr[] = { 1, 2, 3, 4, 5 }; *// 명칭, 초기값 변경 불가*

void main(void)

{

int inx, \*ptr;

for(inx = 1; inx < 5; inx++)

printf("%d \n", arr[inx]);

for(ptr = &arr[1]; ptr <= &arr[4] ; ptr++)

printf("%d \n", \*ptr);

return 0;

}

출력 결과

전역 변수로 arr배열이 0-4로 할당되어 있고

메인 안에 있는 첫 번째 반복문으로 1부터 4까지를 출력하고 있습니다

두 번째 반복문에서는 포인터 변수가 arr[1]~arr[4]의 주소를 참고하고

\*ptr을 통해 주소 안에 있는 값을 출력을 합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명