



---

# R E P O R T

## C프로그래밍2 과제12

과목명	C 프로그래밍 II
분반	2 분반
교수	정 구 철
학번	2020136129
이름	최 수 연
제출일	2020년 11월 25일 수요일

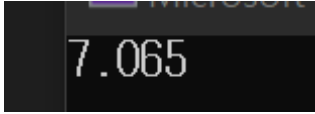
**전처리 및 다중 소스 파일 - 12th**

1. 지름이 3인 원의 넓이를 구하시오

- ☐ 반드시 #define을 사용해서 PI를 정의해야함
- ☐ PI = 3.14

```
#include <stdio.h>
#define PI 3.14

int main(void)
{
    printf("%.3f\n", 1.5 * 1.5 * PI);
    return 0;
}
```



7.065

(2~4). python PORTING 프로젝트,

2. python의 논리 연산자에는 and,or,not이 있다. 이는 C언어에서의 &&, ||, !에 해당되는 것들이다. C언어의 위 세가지 논리 연산자를 #define을 사용하여 python처럼 변경하시오.

그리고 이것들을 이용하여 다음 논리 연산의 결과를 출력하시오.

2-1: ( (1 > 2) and (2 < 3))

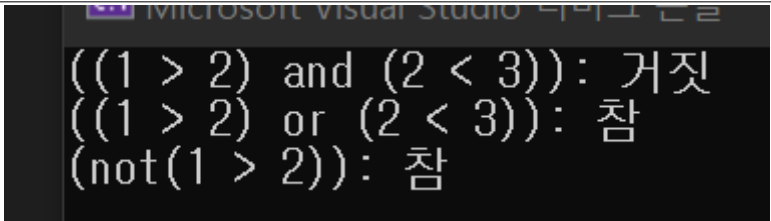
2-2: ( (1 > 2) or (2 < 3))

2-3: (not(1>2))

```
#include <stdio.h>
#define and &&
#define or ||
#define not !

int main(void)
{
    if (((1 > 2) and (2 < 3)) == 1)
        printf("((1 > 2) and (2 < 3)): 참\n");
    else if (((1 > 2) and (2 < 3)) == 0)
        printf("((1 > 2) and (2 < 3)): 거짓\n");
    if (((1 > 2) or (2 < 3)) == 1)
        printf("((1 > 2) or (2 < 3)): 참\n");
    else if (((1 > 2) or (2 < 3)) == 0)
        printf("((1 > 2) or (2 < 3)): 거짓\n");
    if ((not(1 > 2)) == 1)
        printf("(not(1 > 2)): 참\n");
    else if ((not(1 > 2)) == 0)
        printf("(not(1 > 2)): 거짓\n");
}
```

```
printf("(not(1 > 2)): 거짓\n");
return 0;
}
```



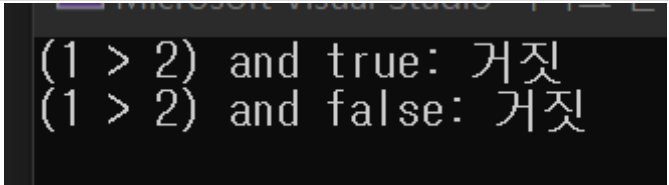
3. python의 true와 false가 있지만, C언어의 가장 기본적인 연산자에는 bool 연산자가 없다. 단순히 '0이 아닌 것'과 '0인 것'으로 true/false를 구분한다. #define을 활용하여 true, false를 만들고 다음 논리연산의 결과를 출력하시오. 2번에서 정의한 and, or, not사용

- 3-1: (1 > 2) and true
- 3-2: (1 > 2) and false

```
#include <stdio.h>
#define and &&
#define or ||
#define not !
#define true 1
#define false 0

int main(void)
{
    if (((1 > 2) and true) == 1)
        printf("(1 > 2) and true: 참\n");
    else if (((1 > 2) and true) == 0)
        printf("(1 > 2) and true: 거짓\n");
    if (((1 > 2) and false) == 1)
        printf("(1 > 2) and false: 참\n");
    else if (((1 > 2) and false) == 0)
        printf("(1 > 2) and false: 거짓\n");

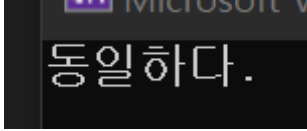
    return 0;
}
```



4. python에선 ==을 is로 사용한다. is를 사용하여 'a'와 'a'가 같은지 확인하시오

```
#include <stdio.h>
#define is ==

int main(void)
{
    char a = 'a';
    if (a is 'a')
        printf("동일하다.\n");
    else
        printf("동일하지 않다.\n");
    return 0;
}
```

A terminal window with a black background and white text. The text "동일하다." is displayed on a single line.

5. 프로그램을 디버깅할 때 자주 사용되는 ASSERT 매크로를 작성해보자. int형 변수 a= 1이며 ASSERT 값은 a=1이 아닐 때 printf("source file is fail")을 출력한다.

```
#include <stdio.h>
#include <stdlib.h>
#define ASSERT(x) {if (!(x)){ printf("source file is fail\n",__FILE__, __LINE__), exit(1);}}

int main(void)
{
    int a = 1;
    ASSERT(a != 1);
    return 0;
}
```

A terminal window with a black background and white text. The text "source file is fail" is displayed on a single line.

6. 프로그램을 디버깅할 때 자주 사용되는 ASSERT 매크로를 작성해보자. 이때 오류가 난 행과 파일이름을 출력할 수 있어야한다. int형 변수 a= 1이며 ASSERT 값은 a=1이 아닐 때 printf("source file is fail")을 출력한다.

```
#include <stdio.h>
#include <stdlib.h>
#define ASSERT(x) {if (!(x))\
{ printf("source file is fail\n소스파일 %s의 %d번째 줄에서 실패\n",__FILE__,\
__LINE__),\
exit(1);}}

int main(void)
{
    int a = 1;
    ASSERT(a != 1);
    return 0;
}
```

```
source file is fail
소스파일 C:\Users\choi6\source\repos\Project4\Project4\소스.cpp의 10번째 줄에서 실패
```

7. 둘 중에 어떤 것이 큰지 반환하는 MAX함수를 #define을 통해 정의하시오. 실험은 int a=1, b=2로 실험하시오

```
#include <stdio.h>
#define MAX(x, y) ((x > y) ? x : y)

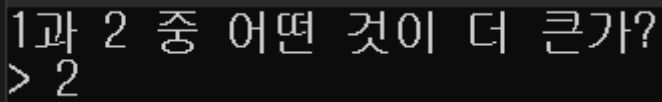
int main(void)
{
    int a = 1, b = 2;
    printf("1과 2 중 어떤 것이 더 큰가?\n> %d\n", MAX(a, b));
    return 0;
}
```

```
1과 2 중 어떤 것이 더 큰가?
> 2
```

8. 둘 중에 어떤 것이 큰지 반환하는 MAX함수를 #define을 통해 정의하시오. 실험은 double a=1, b=2로 실험하시오

```
#include <stdio.h>
#define MAX(x, y) ((x > y) ? x : y)

int main(void)
{
    double a = 1, b = 2;
    printf("1과 2 중 어떤 것이 더 큰가?\n> %.0f\n", MAX(a, b));
    return 0;
}
```

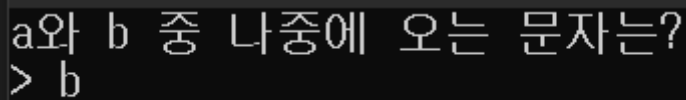


```
1과 2 중 어떤 것이 더 큰가?
> 2
```

9. 아스키코드에서 둘 중에 어떤 것이 나중에 오는지를 반환하는 LATTER함수를 #define을 통해 정의하시오. 실험은 char a='a', b='b'로 실험하시오

```
#include <stdio.h>
#define LATTER(x, y) ((x > y) ? x : y)

int main(void)
{
    char a = 'a', b = 'b';
    printf("a와 b 중 나중에 오는 문자는?\n> %c\n", LATTER(a, b));
    return 0;
}
```



```
a와 b 중 나중에 오는 문자는?
> b
```

10. 어떤 타입의 수라도 절대값으로 만드는 ABS함수를 #define을 통해 정의하시오. 실험은 int a=-2, double b= -3.23로 실험하시오.

```
#include <stdio.h>
#define ABS(x) ((x) > 0 ? (x) : -(x))

int main(void)
{
    int a = -2;
    double b = -3.23;
    printf("a의 절댓값: %d, b의 절댓값: %.2f\n", ABS(a), ABS(b));
    return 0;
}
```

a의 절댓값: 2, b의 절댓값: 3.23