

7번문제.

```
int mult2(int num)
{
    return num * 2;
}

int main(void)
{
    int i, sum = 0, result;
    for(i = 0; i < 5; i++)
        sum += i;
    result = mult2(sum);
    return 0;
}
```

| | | |
|------------------------------|------------------------|------------------------|
| 0x00000000004004e4 <+0>: | push %rbp | -->스택만들기 |
| 0x00000000004004e5 <+1>: | mov %rsp,%rbp | -->스택베이스포인터설정 |
| 0x00000000004004e8 <+4>: | sub \$0x10,%rsp | -->0x10 만큼 스택공간할당 |
| 0x00000000004004ec <+8>: | movl \$0x0,-0x8(%rbp) | -->기준점에서 0x8,0xc 떨어진 |
| 0x00000000004004f3 <+15>: | movl \$0x0,-0xc(%rbp) | 지점에 0 을 넣음 |
| 0x00000000004004fa <+22>: | jmp 0x400506 <main+34> | -->34 만큼떨어진곳으로 점프 |
| 0x00000000004004fc <+24>: | mov -0xc(%rbp),%eax | -->베이스포인터에서 0xc 떨어진 |
| | | 곳에있는값을 eax 에복사 |
| 0x00000000004004ff <+27>: | add %eax,-0x8(%rbp) | -->eax 와-0x8 안에 값을 더해 |
| | | 서 -0x8 에다 넣음 |
| 0x0000000000400502 <+30>: | addl \$0x1,-0xc(%rbp) | -->1 을-0xc 에다 더해넣음 |
| 0x0000000000400506 <+34>: | cmpl \$0x4,-0xc(%rbp) | -->0x4 와 0xc 지점에있는걸비교 |
| 0x000000000040050a <+38>: | jle 0x4004fc <main+24> | -->작거나같으면점프 |
| 0x000000000040050c <+40>: | mov -0x8(%rbp),%eax | -->-0x8 지점에있는걸 eax 에복사 |
| 0x000000000040050f <+43>: | mov %eax,%edi | -->eax 의값을 edi 에복사 |
| => 0x0000000000400511 <+45>: | callq 0x4004d6 <mult2> | -->함수호출 mult2 |
| 0x0000000000400516 <+50>: | mov %eax,-0x4(%rbp) | -->eax 의값을-0x4 지점에복사 |
| 0x0000000000400519 <+53>: | mov \$0x0,%eax | -->eax 를 0 으로만듬 |
| 0x000000000040051e <+58>: | leaveq | -->떠나고 |
| 0x000000000040051f <+59>: | retq | ->반환? q 는 qword |
| ----- | | |
| => 0x00000000004004d6 <+0>: | push %rbp | -->스택할당 |
| 0x00000000004004d7 <+1>: | mov %rsp,%rbp | -->기준점 |
| 0x00000000004004da <+4>: | mov %edi,-0x4(%rbp) | -->기준점 0x4 지점에 edi 복사 |

| | | | |
|----------------------------|------|-----------------|-------------------------|
| 0x000000000004004dd <+7>: | mov | -0x4(%rbp),%eax | -->0x4 지점에있는걸 eax 에복사 |
| 0x000000000004004e0 <+10>: | add | %eax,%eax | -->eax 에있는걸 더해서 eax 에더함 |
| 0x000000000004004e2 <+12>: | pop | %rbp | -->rbp 에있는값을꺼냄 |
| 0x000000000004004e3 <+13>: | retq | | -->반환 |

12 번문제->

Visual Studio 에서 나오는 디버깅은 어셈블리어코드를 보여주지 않는것 같다.

리눅스에있는 디버거와 비슷한 점은 한줄한줄 실행해서 오류가 난 지점을 파악할 수 있게 하는 것 같다.

디버깅 시작은 F5 를 누르면 시작할 수 있고 쉬프트 F5 를 누르면 디버깅이 중단된다.

단축기가 아니어도 VisualStudio 좌측 상단에 메뉴에 디버그라는 메뉴가 있다

프로세스에 연결, 예외, 한 단계씩 코드 실행 F11, 프로시저 단위 실행 F10

중단점 설정/해제 F9 새중단점

모든 중단점 삭제(D)Ctrl+Shift+F9

모든 중단점 해제(N) 등이 있ㄷㅏ.

디버깅 시작은 말그대로 디버깅 시작하는 단축키이고 중단은 중단하는거다

한 단계씩 코드실행은 프로그램을 한줄씩 실행시켜서 프로그램이 어디서 문제가 있는지 확인할 수 있다.

프로시저단위는 문장단위로 실행하게 된다 함수를 만났을때도 무조건 한줄 단위로 실행하므로 함수 내부로는 진입하지 않게 된다.

중단점 사이가 너무 멀면 F5 를 사용해서 바로 다음 중단점으로 이동할 수 있다.