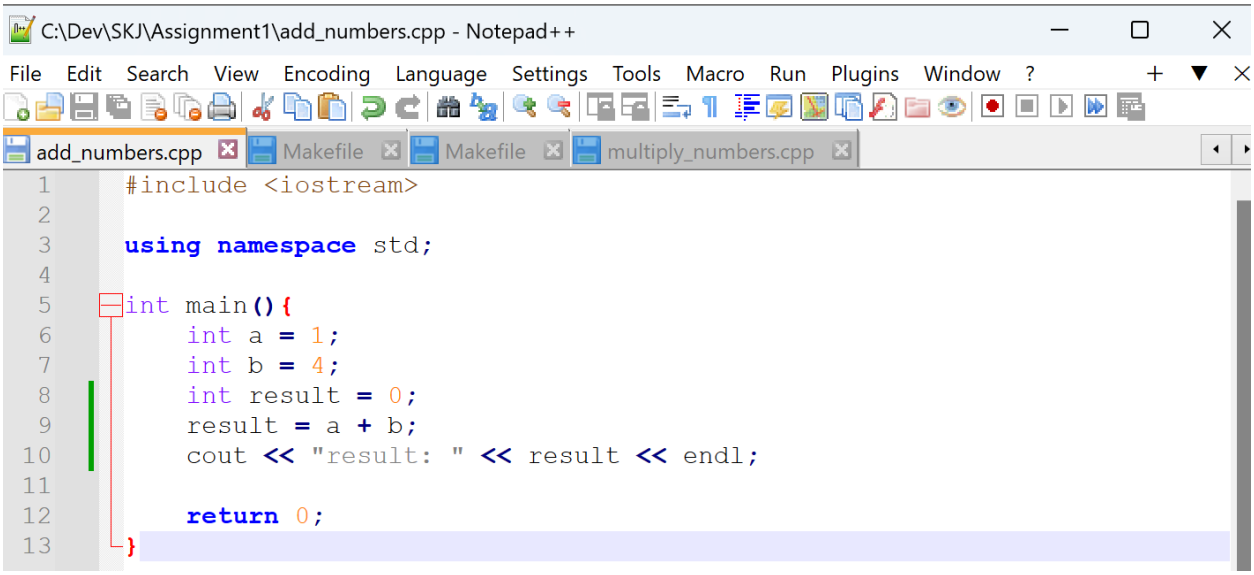
Nama : Kosmas Rio Legowo

NIM : 23/512012/PA/21863

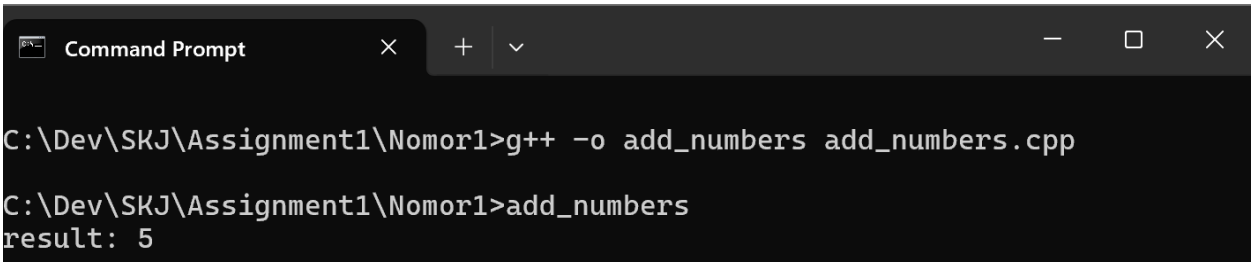
GitHub link : https://github.com/kosmasrio0411/Kosmas-Rio-Legowo-SKJ-Lab.git

1.6.2

1. C++ program to add two integers.



2. Compiling the code



3. Disassemble the Code

```
C:\Dev\SKJ\Assignment1\Nomor1>objdump -d add_numbers.exe

add_numbers.exe:     file format pei-x86-64


Disassembly of section .text:

0000000140001000 <__mingw_invalidParameterHandler>:
   140001000:   c3                      ret
   140001001:   66 66 2e 0f 1f 84 00    data16 cs nopw 0x0(%rax,%rax,1)
   140001008:   00 00 00 00
   14000100c:   0f 1f 40 00             nopl   0x0(%rax)

0000000140001010 <pre_c_init>:
   140001010:   48 83 ec 28             sub    $0x28,%rsp
   140001014:   48 8b 05 15 34 00 00    mov    0x3415(%rip),%rax        # 140004430 <.refptr.__mingw_initltsdrot_force>
   14000101b:   31 c9                   xor    %ecx,%ecx
   14000101d:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   140001023:   48 8b 05 16 34 00 00    mov    0x3416(%rip),%rax        # 140004440 <.refptr.__mingw_initltsdyn_force>
   14000102a:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   140001030:   48 8b 05 19 34 00 00    mov    0x3419(%rip),%rax        # 140004450 <.refptr.__mingw_initltssuo_force>
   140001037:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   14000103d:   48 8b 05 6c 33 00 00    mov    0x336c(%rip),%rax        # 1400043b0 <.refptr.__ImageBase>
   140001044:   66 81 38 4d 5a          cmpw   $0x5a4d,(%rax)
   140001049:   75 0f                   jne    14000105a <pre_c_init+0x4a>
   14000104b:   48 63 50 3c             movslq 0x3c(%rax),%rdx
   14000104f:   48 01 d0                add    %rdx,%rax
   140001052:   81 38 50 45 00 00       cmpl   $0x4550,(%rax)
   140001058:   74 66                   je     1400010c0 <pre_c_init+0xb0>
   14000105a:   48 8b 05 bf 33 00 00    mov    0x33bf(%rip),%rax        # 140004420 <.refptr.__mingw_app_type>
   140001061:   89 0d a5 5f 00 00       mov    %ecx,0x5fa5(%rip)        # 14000700c <managedapp>
   140001067:   8b 00                   mov    (%rax),%eax
   140001069:   85 c0                   test   %eax,%eax
   14000106b:   74 43                   je     1400010b0 <pre_c_init+0xa0>
   14000106d:   b9 02 00 00 00          mov    $0x2,%ecx
   140001072:   e8 a1 15 00 00          call   140002618 <__set_app_type>
   140001077:   e8 24 15 00 00          call   1400025a0 <__p__fmode>
   14000107c:   48 8b 15 6d 34 00 00    mov    0x346d(%rip),%rdx        # 1400044f0 <.refptr._fmode>
   140001083:   8b 12                   mov    (%rdx),%edx
   140001085:   89 10                   mov    %edx,(%rax)
   140001087:   e8 24 15 00 00          call   1400025b0 <__p__commode>
   14000108c:   48 8b 15 3d 34 00 00    mov    0x343d(%rip),%rdx        # 1400044d0 <.refptr._commode>
   140001093:   8b 12                   mov    (%rdx),%edx
   140001095:   89 10                   mov    %edx,(%rax)
   140001097:   e8 14 05 00 00          call   1400015b0 <_setargv>
   14000109c:   48 8b 05 cd 32 00 00    mov    0x32cd(%rip),%rax        # 140004370 <.refptr._MINGW_INSTALL_DEBUG_MATHERR>
   1400010a3:   83 38 01                cmpl   $0x1,(%rax)
   1400010a6:   74 50                   je     1400010f8 <pre_c_init+0xe8>
   1400010a8:   31 c0                   xor    %eax,%eax
   1400010aa:   48 83 c4 28             add    $0x28,%rsp
   1400010ae:   c3                      ret
   1400010af:   90                      nop
   1400010b0:   b9 01 00 00 00          mov    $0x1,%ecx
   1400010b5:   e8 5e 15 00 00          call   140002618 <__set_app_type>
   1400010ba:   eb bb                   jmp    140001077 <pre_c_init+0x67>
   1400010bc:   0f 1f 40 00             nopl   0x0(%rax)
   1400010c0:   0f b7 50 18             movzwl 0x18(%rax),%edx
   1400010c4:   66 81 fa 0b 01          cmp    $0x10b,%dx
   1400010c9:   74 45                   je     140001110 <pre_c_init+0x100>
   1400010cb:   66 81 fa 0b 02          cmp    $0x20b,%dx
   1400010d0:   75 88                   jne    14000105a <pre_c_init+0x4a>
   1400010d2:   83 b8 84 00 00 00 0e    cmpl   $0xe,0x84(%rax)
   1400010d9:   0f 86 7b ff ff ff       jbe    14000105a <pre_c_init+0x4a>
   1400010df:   8b 90 f8 00 00 00       mov    0xf8(%rax),%edx
   1400010e5:   31 c9                   xor    %ecx,%ecx
   1400010e7:   85 d2                   test   %edx,%edx
   1400010e9:   0f 95 c1                setne  %cl
   1400010ec:   e9 69 ff ff ff          jmp    14000105a <pre_c_init+0x4a>
   1400010f1:   0f 1f 80 00 00 00 00    nopl   0x0(%rax)
   1400010f8:   48 8b 0d 11 34 00 00    mov    0x3411(%rip),%rcx        # 140004510 <.refptr._matherr>
   1400010ff:   e8 1c 0c 00 00          call   140001d20 <__mingw_setusermatherr>
```

main

```
0000000140001450 <main>:
   140001450:   55                      push   %rbp
   140001451:   48 89 e5                mov    %rsp,%rbp
   140001454:   48 83 ec 30             sub    $0x30,%rsp
   140001458:   e8 33 01 00 00          call   140001590 <__main>
   14000145d:   c7 45 fc 01 00 00 00    movl   $0x1,-0x4(%rbp)
   140001464:   c7 45 f8 04 00 00 00    movl   $0x4,-0x8(%rbp)
   14000146b:   c7 45 f4 00 00 00 00    movl   $0x0,-0xc(%rbp)
   140001472:   8b 55 fc                mov    -0x4(%rbp),%edx
   140001475:   8b 45 f8                mov    -0x8(%rbp),%eax
   140001478:   01 d0                   add    %edx,%eax
   14000147a:   89 45 f4                mov    %eax,-0xc(%rbp)
   14000147d:   48 8d 05 7c 2b 00 00    lea    0x2b7c(%rip),%rax        # 140004000 <.rdata>
   140001484:   48 89 c2                mov    %rax,%rdx
   140001487:   48 8b 05 f2 2e 00 00    mov    0x2ef2(%rip),%rax        # 140004380 <__fu0__ZSt4cout>
   14000148e:   48 89 c1                mov    %rax,%rcx
   140001491:   e8 2a 00 00 00          call   1400014c0 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc>
   140001496:   48 89 c1                mov    %rax,%rcx
   140001499:   8b 45 f4                mov    -0xc(%rbp),%eax
   14000149c:   89 c2                   mov    %eax,%edx
   14000149e:   e8 2d 00 00 00          call   1400014d0 <_ZNSolsEi>
   1400014a3:   48 89 c1                mov    %rax,%rcx
   1400014a6:   48 8b 05 e3 2e 00 00    mov    0x2ee3(%rip),%rax        # 140004390 <.refptr._ZSt4endlIcSt11char_traitsIcEER
St13basic_ostreamIT_T0_ES6_>
   1400014ad:   48 89 c2                mov    %rax,%rdx
   1400014b0:   e8 23 00 00 00          call   1400014d8 <_ZNSolsEPFRSoS_E>
   1400014b5:   b8 00 00 00 00          mov    $0x0,%eax
   1400014ba:   48 83 c4 30             add    $0x30,%rsp
   1400014be:   5d                      pop    %rbp
   1400014bf:   c3                      ret
```

4. Write and run Makefile

C:\Dev\SKJ\Assignment1\Nomor2\Makefile - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

add_numbers.cpp | Makefile | Makefile | multiply_numbers.cpp

```
1  all: multiply_numbers
2
3  multiply_numbers: multiply_numbers.cpp
4      g++ -o multiply_numbers multiply_numbers.cpp
5
6  dump: multiply_numbers
7      objdump -d multiply_numbers.exe > multiply_numbers.asm
8
9  clean:
10     rm -f multiply_numbers multiply_numbers.asm
11
12 run: multiply_numbers
13     ./multiply_numbers
```

add_numbers.asm

File Edit View

```
add_numbers.exe:     file format pei-x86-64


Disassembly of section .text:

0000000140001000 <__mingw_invalidParameterHandler>:
   140001000:   c3                      ret
   140001001:   66 66 2e 0f 1f 84 00    data16 cs nopw 0x0(%rax,%rax,1)
   140001008:   00 00 00 00
   14000100c:   0f 1f 40 00             nopl   0x0(%rax)

0000000140001010 <pre_c_init>:
   140001010:   48 83 ec 28             sub    $0x28,%rsp
   140001014:   48 8b 05 15 34 00 00    mov    0x3415(%rip),%rax        # 140004430
<.refptr.__mingw_initltsdrot_force>
   14000101b:   31 c9                   xor    %ecx,%ecx
   14000101d:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   140001023:   48 8b 05 16 34 00 00    mov    0x3416(%rip),%rax        # 140004440
<.refptr.__mingw_initltsdyn_force>
   14000102a:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   140001030:   48 8b 05 19 34 00 00    mov    0x3419(%rip),%rax        # 140004450
<.refptr.__mingw_initltssuo_force>
   140001037:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   14000103d:   48 8b 05 6c 33 00 00    mov    0x336c(%rip),%rax        # 1400043b0
<.refptr.__ImageBase>
   140001044:   66 81 38 4d 5a          cmpw   $0x5a4d,(%rax)
   140001049:   75 0f                   jne    14000105a <pre_c_init+0x4a>
   14000104b:   48 63 50 3c             movslq 0x3c(%rax),%rdx
   14000104f:   48 01 d0                add    %rdx,%rax
   140001052:   81 38 50 45 00 00       cmpl   $0x4550,(%rax)
   140001058:   74 66                   je     1400010c0 <pre_c_init+0xb0>
   14000105a:   48 8b 05 bf 33 00 00    mov    0x33bf(%rip),%rax        # 140004420
<.refptr.__mingw_app_type>
   140001061:   89 0d a5 5f 00 00       mov    %ecx,0x5fa5(%rip)        # 14000700c
<managedapp>
   140001067:   8b 00                   mov    (%rax),%eax
   140001069:   85 c0                   test   %eax,%eax
   14000106b:   74 43                   je     1400010b0 <pre_c_init+0xa0>
   14000106d:   b9 02 00 00 00          mov    $0x2,%ecx
   140001072:   e8 a1 15 00 00          call   140002618 <__set_app_type>
   140001077:   e8 24 15 00 00          call   1400025a0 <__p__fmode>
   14000107c:   48 8b 15 6d 34 00 00    mov    0x346d(%rip),%rdx        # 1400044f0
```

Ln 1, Col 1 | 107,330 characters | 100% | Windows (CRLF) | UTF-8

Command Prompt

```
C:\Dev\SKJ\Assignment1\Nomor1>make all
g++ -o add_numbers add_numbers.cpp

C:\Dev\SKJ\Assignment1\Nomor1>make add_numbers
g++ -o add_numbers add_numbers.cpp

C:\Dev\SKJ\Assignment1\Nomor1>make dump
g++ -o add_numbers add_numbers.cpp
objdump -d add_numbers.exe > add_numbers.asm

C:\Dev\SKJ\Assignment1\Nomor1>add_numbers.asm

C:\Dev\SKJ\Assignment1\Nomor1>make clean
rm -f add_numbers add_numbers.asm

C:\Dev\SKJ\Assignment1\Nomor1>add_numbers.asm
'add_numbers.asm' is not recognized as an internal or external command,
operable program or batch file.

C:\Dev\SKJ\Assignment1\Nomor1>make run
g++ -o add_numbers add_numbers.cpp
./add_numbers
result: 5

C:\Dev\SKJ\Assignment1\Nomor1>
```

1.6.3

1. Analyzing the Assembly Code

```
section .data
```

```
    num1 dw 5
```

```
    num2 dw 10
```

```
    result dw 0
```

Section .data adalah di mana data (variabel) untuk program tersebut didefinisikan. Baris `num1 dw 5` mendefinisikan num1 sebagai word (dw adalah define word (yang ukurannya 16 bit)) dengan value 5. Baris berikutnya juga mendefinisikan yaitu num2 sebagai word dengan value 10. Kemudian baris berikutnya mendefinisikan result sebagai word dengan value 0.

```
section .text
```

```
    global _start
```

Section .text adalah tempat di mana instruksi untuk program yang ditulis. Kemudian global _start mendeklarasikan _start sebagai simbol global, yang berarti ini adalah entry point suatu program. Hal ini menjadi penting agar sistem operasi mengetahui di mana untuk mulai eksekusi program.

```
_start:
```

_start seperti sudah disebutkan di atas, adalah titik awal program.

```
    mov ax, [num1]
```

Instruksi ini memindahkan nilai num1 (5) ke dalam register AX. Sekarang AX berisi nilai 5.

```
    imul ax, [num2]
```

Instruksi ini mengalikan nilai num2 (10) dengan nilai yang sekarang ada di dalam register AX (5). Setelah operasi tersebut, register AX akan berisi hasil dari 5 * 10 yaitu 50.

```
    mov [result], ax
```

Instruksi ini menyimpan hasil perkalian yang saat ini ada di dalam register AX ke variabel result. Jadi, variabel result sekarang memiliki nilai 50.

```
    ; Exit the program
```

Berikut adalah instruksi-instruksi untuk mengakhiri program

```
mov eax, 1
```

Instruksi ini memindahkan nilai 1 ke dalam register EAX. Dalam system call Linux, 1 adalah kode untuk system call ‘exit’. Program akan berhenti ketika system call ini dipanggil.

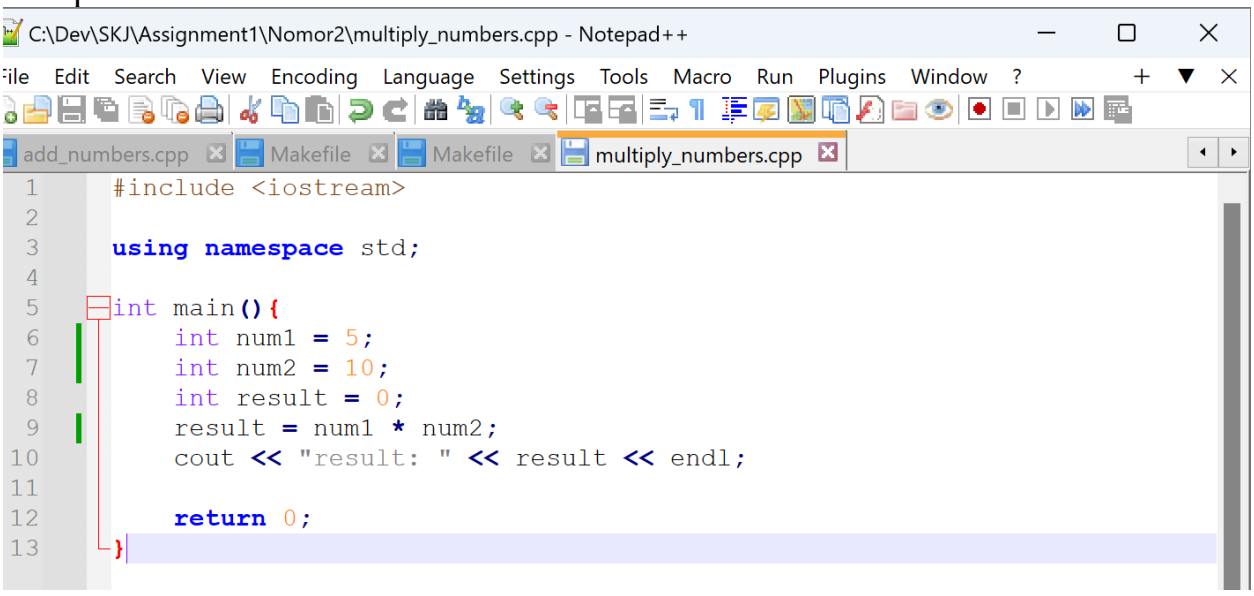
```
xor ebx, ebx
```

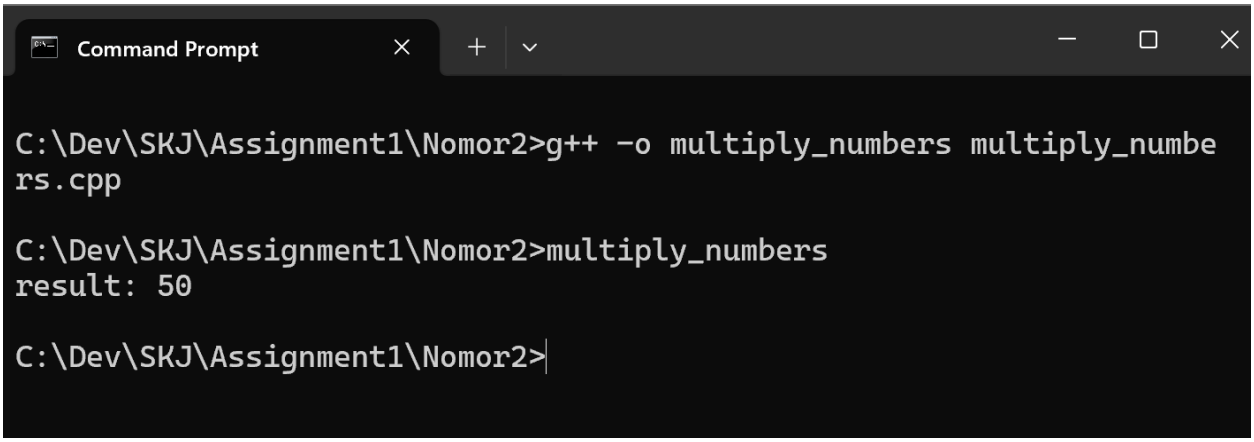
Instruksi ini mengosongkan register EBX dengan melakukan operasi XOR antara EBX dengan EBX (dirinya sendiri) sehingga nilainya menjadi 0. Ini adalah exit status dari sebuah program yang menunjukkan normal exit tanpa error.

```
int 0x80
```

Ini memicu interrupt, memanggil kernel untuk menangani system call. Karena EAX bernilai 1, sistem mengetahui ini sebagai permintaan untuk exit program dan status keluarnya 0 (nilai di EBX).

2. Equivalent c++ Code





3. Write and Run Makefile

C:\Dev\SKJ\Assignment1\Nomor2\Makefile - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

add_numbers.cpp | Makefile | Makefile | multiply_numbers.cpp

```
1  all: multiply_numbers
2
3  multiply_numbers: multiply_numbers.cpp
4      g++ -o multiply_numbers multiply_numbers.cpp
5
6  dump: multiply_numbers
7      objdump -d multiply_numbers.exe > multiply_numbers.asm
8
9  clean:
10     rm -f multiply_numbers multiply_numbers.asm
11
12 run: multiply_numbers
13     ./multiply_numbers
```

add_numbers.asm | multiply_numbers.asm

File Edit View

```
multiply_numbers.exe:     file format pei-x86-64


Disassembly of section .text:

0000000140001000 <__mingw_invalidParameterHandler>:
   140001000:   c3                      ret
   140001001:   66 66 2e 0f 1f 84 00    data16 cs nopw 0x0(%rax,%rax,1)
   140001008:   00 00 00 00
   14000100c:   0f 1f 40 00             nopl   0x0(%rax)

0000000140001010 <pre_c_init>:
   140001010:   48 83 ec 28             sub    $0x28,%rsp
   140001014:   48 8b 05 15 34 00 00    mov    0x3415(%rip),%rax        # 140004430
<.refptr.__mingw_initltsdrot_force>
   14000101b:   31 c9                   xor    %ecx,%ecx
   14000101d:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   140001023:   48 8b 05 16 34 00 00    mov    0x3416(%rip),%rax        # 140004440
<.refptr.__mingw_initltsdyn_force>
   14000102a:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   140001030:   48 8b 05 19 34 00 00    mov    0x3419(%rip),%rax        # 140004450
<.refptr.__mingw_initltssuo_force>
   140001037:   c7 00 01 00 00 00       movl   $0x1,(%rax)
   14000103d:   48 8b 05 6c 33 00 00    mov    0x336c(%rip),%rax        # 1400043b0
<.refptr.__ImageBase>
   140001044:   66 81 38 4d 5a          cmpw   $0x5a4d,(%rax)
   140001049:   75 0f                   jne    14000105a <pre_c_init+0x4a>
   14000104b:   48 63 50 3c             movslq 0x3c(%rax),%rdx
   14000104f:   48 01 d0                add    %rdx,%rax
   140001052:   81 38 50 45 00 00       cmpl   $0x4550,(%rax)
   140001058:   74 66                   je     1400010c0 <pre_c_init+0xb0>
   14000105a:   48 8b 05 bf 33 00 00    mov    0x33bf(%rip),%rax        # 140004420
<.refptr.__mingw_app_type>
   140001061:   89 0d a5 5f 00 00       mov    %ecx,0x5fa5(%rip)        # 14000700c
<managedapp>
   140001067:   8b 00                   mov    (%rax),%eax
   140001069:   85 c0                   test   %eax,%eax
   14000106b:   74 43                   je     1400010b0 <pre_c_init+0xa0>
   14000106d:   b9 02 00 00 00          mov    $0x2,%ecx
   140001072:   e8 a1 15 00 00          call   140002618 <__set_app_type>
   140001077:   e8 24 15 00 00          call   1400025a0 <__p__fmode>
   14000107c:   48 8b 15 6d 34 00 00    mov    0x346d(%rip),%rdx        # 1400044f0
```

Ln 1, Col 1 | 107,322 characters | 100% | Windows (CRLF) | UTF-8

Command Prompt

```
C:\Dev\SKJ\Assignment1\Nomor2>make all
g++ -o multiply_numbers multiply_numbers.cpp

C:\Dev\SKJ\Assignment1\Nomor2>make run
g++ -o multiply_numbers multiply_numbers.cpp
./multiply_numbers
result: 50

C:\Dev\SKJ\Assignment1\Nomor2>make dump
g++ -o multiply_numbers multiply_numbers.cpp
objdump -d multiply_numbers.exe > multiply_numbers.asm

C:\Dev\SKJ\Assignment1\Nomor2>multiply_numbers.asm

C:\Dev\SKJ\Assignment1\Nomor2>make clear
make: *** No rule to make target 'clear'.  Stop.

C:\Dev\SKJ\Assignment1\Nomor2>make clean
rm -f multiply_numbers multiply_numbers.asm

C:\Dev\SKJ\Assignment1\Nomor2>multiply_numbers.asm
'multiply_numbers.asm' is not recognized as an internal or external
command,
operable program or batch file.

C:\Dev\SKJ\Assignment1\Nomor2>make run
g++ -o multiply_numbers multiply_numbers.cpp
./multiply_numbers
result: 50

C:\Dev\SKJ\Assignment1\Nomor2>
```