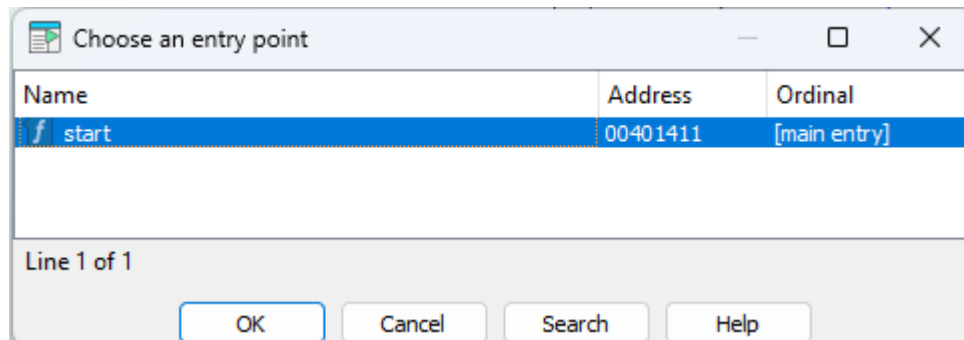


Question 1

1. Sections of the program:

Name	Start	End	R	W	X
.text	00401000	00402000	R	.	X
.idata	00402000	004020D0	R	.	.
.rdata	004020D0	00403000	R	.	.
.data	00403000	00404000	R	W	.

2. entry point:



3. Main function:



4. The next questions 4-8 will be answered together as a story:

First error:

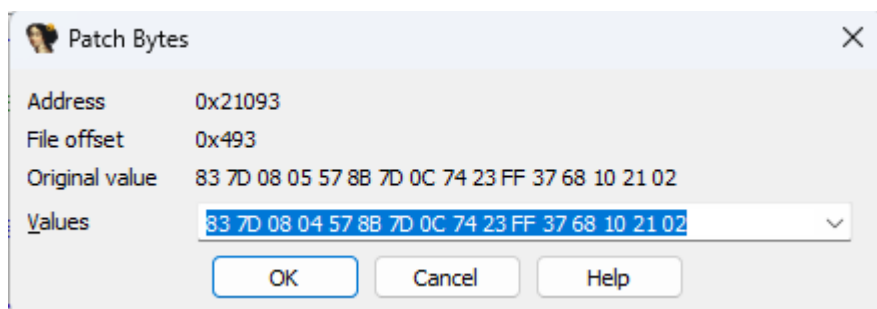
Wrong number of arguments are checked.

We try to run the program as the Usage says:

```
C:\Users\user\Desktop\hw3-samples\samples>.\q1.exe "C:\Users\user\Desktop\hw3-samples\samples\q2.exe" 1 20
Usage: .\q1.exe <file_path> <min_length> <max_length>
```

However it still shows the usage, in the assembly code we can see that cmp is made with the number 5, however there are only 4 arguments mentioned in the Usage.

```
.text:00021090      mov     [ebp+var_4], eax
.text:00021093      cmp     [ebp+argc], 5 ; If argc equals 5, it branches to a specific location (loc_4010C0) for further instructions.
.text:00021097      nush    edi
```



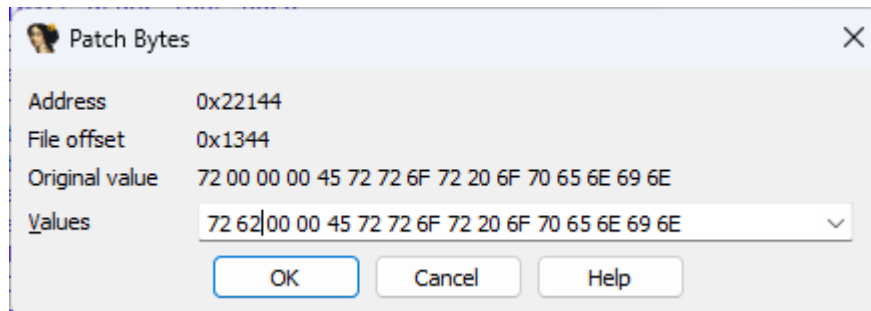
After patching the value from 5 to 4 we will get past the first error!
Success!!

Let's view the next problem:

```
C:\Users\user\Desktop\hw3-samples\samples>.\q1.exe "C:\Users\user\Desktop\hw3-samples\samples\q2.exe" 1 20
Error opening file C:\Users\user\Desktop\hw3-samples\samples\q2.exe
```

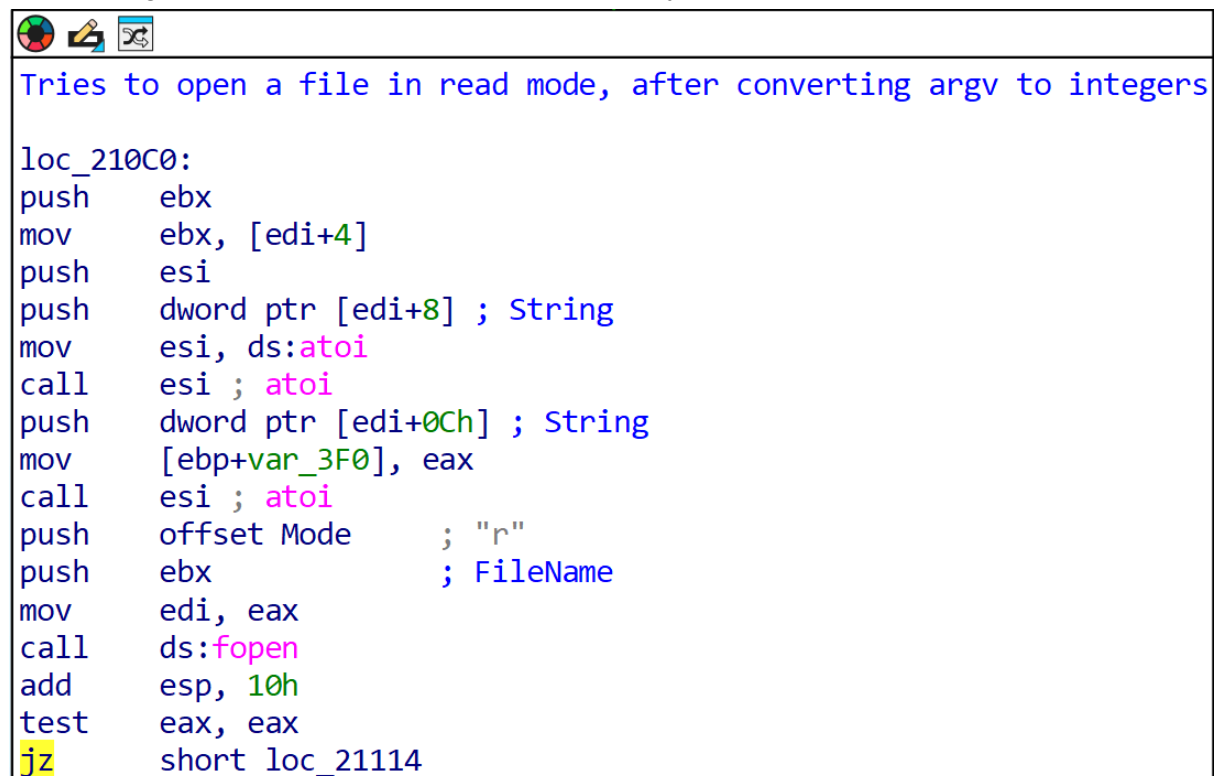
We see that the file is open for 'r' - reading but not in 'rb' so I tried to patch it.

```
.rdata:00022144 Mode db 'r',0 ; DATA XREF: _main+5B↑to
```



no success, appears that it's not the problem - or just part of it.

let's look again on the code snippet which should try to open a file:

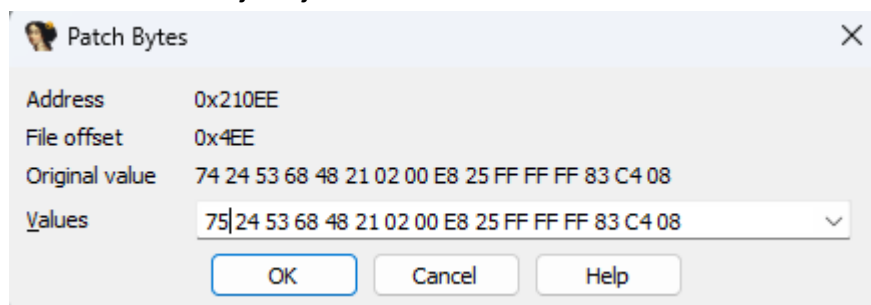


We notice that after call to fopen there is test eax, eax.

If fopen fails it will return 0 (stored in eax) and the test operation will set the zero flag on if the fopen failed.

So jz will jump if the fopen failed. HOWEVER, this wasn't intended, because the code in loc_21114 is handling the case of a successful operation.

We should switch jz to jnz:



After the patch we don't see any errors in the file opening.
Success!

```
C:\Users\user\Desktop\hw3-samples\samples>.\q1.exe "C:\Users\user\Desktop\hw3-samples\samples\q2.exe" 1 20
C:\Users\user\Desktop\hw3-samples\samples>
```

However we don't see the any output, for whatever reason 😞

Let's continue our investigation:

```
; Attributes: bp-based frame

; int __cdecl FormattedInputWrapper(FILE *Stream, char *Format, char Arglist)
FormattedInputWrapper proc near

Stream= dword ptr  8
Format= dword ptr  0Ch
Arglist= byte ptr  10h

push    ebp
mov     ebp, esp
mov     eax, [ebp+Stream]
lea     ecx, [ebp+Arglist]
push    ecx                ; Arglist
push    0                  ; Locale
push    [ebp+Format]       ; Format
push    eax                ; Stream
call    sub_21010
push    dword ptr [eax+4]
push    dword ptr [eax] ; Options
call    ds:__stdio_common_vfscanf
add     esp, 18h
pop     ebp
retn
FormattedInputWrapper endp
```

We note a wrapper to fscanf function, which class sub_21010:

```
sub_21010 proc near
mov     eax, offset unk_23370
retn
sub_21010 endp
```


Which by itself is doing something weird with unk_23370 variable...

Let's try to remove this function from the code with NOPs.

It didn't do much. so let's step back, maybe something else is off.

```
loc_401114:
lea     eax, [ebp+Arglist]
xor     esi, esi
push    eax                ; Arglist
push    offset aS_0        ; "%S"
push    esi                ; Stream
call    FormattedInputWrapper
add     esp, 0Ch
cmp     eax, 0FFFFFFFFh
jz      short loc_40119B
```

It seems like the esi value is 0 (xor esi, esi) and it's pushed as the stream - that's an error -> it should be the file descriptor which is at eax from the call to fopen!!
we need to change the order of the instructions here:

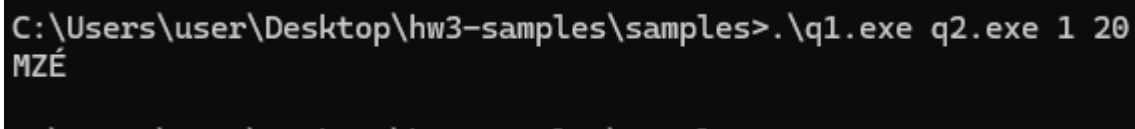


```

loc_401114:
mov     esi, eax
lea     eax, [ebp+ArgList]
push    eax                ; Arglist
push    offset aS_0        ; "%s"
push    esi                ; Stream
call    FormattedInputWrapper
add     esp, 0Ch
cmp     eax, 0FFFFFFFFh
jz      short loc_40119B

```

Now the esi will hold the file descriptor from fopen call. and the rest is unchanged!
Let's try to run the program now!

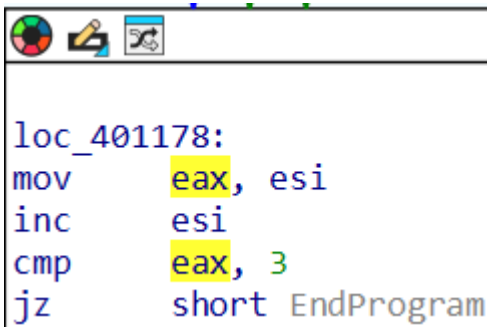


```

C:\Users\user\Desktop\hw3-samples\samples>.\q1.exe q2.exe 1 20
MZÉ

```

it appears that we print only one string...

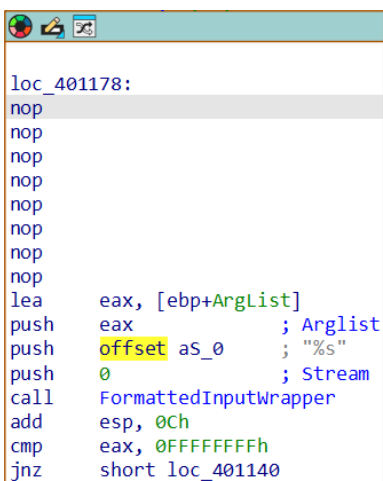


```

loc_401178:
mov     eax, esi
inc     esi
cmp     eax, 3
jz      short EndProgram

```

This code appears to count to 3, and jump to EndProgram - I think it's not useful here also because in esi we hold our file descriptor so incrementing it won't help. Let's patch it with nops:

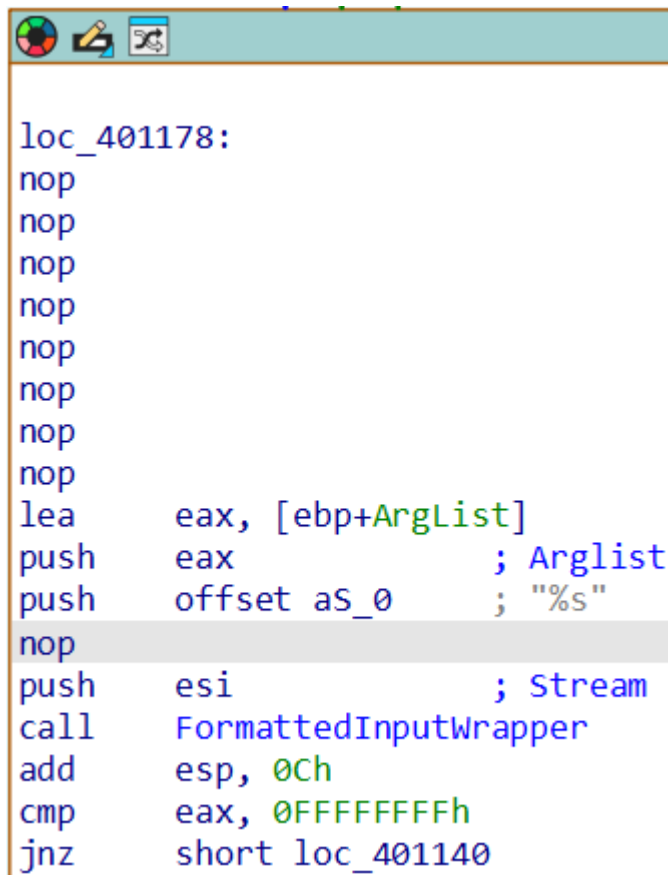
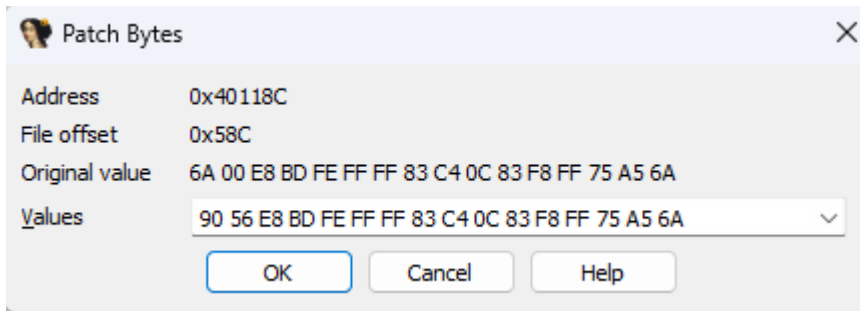


```

loc_401178:
nop
nop
nop
nop
nop
nop
nop
nop
nop
lea     eax, [ebp+ArgList]
push    eax                ; Arglist
push    offset aS_0        ; "%s"
push    0                  ; Stream
call    FormattedInputWrapper
add     esp, 0Ch
cmp     eax, 0FFFFFFFFh
jnz     short loc_401140

```

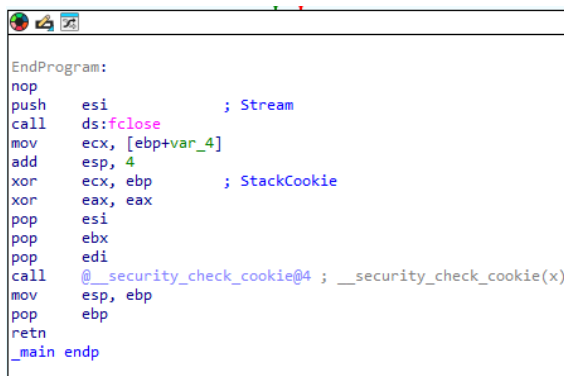
Now we see that the Stream again is equal to 0 and not the file descriptor, let's change it back to push esi.



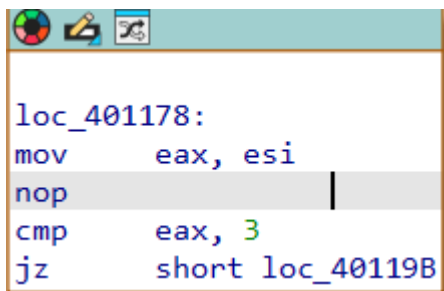
```

C:\Users\user\Desktop\hw3-samples\samples>.\q1.exe q2.exe 1 20
MZÉ
  
```

Last thing we need to fix is the end of program part:
It's pretty simple just push the esi as we already did:



After some testing it appears that we just need to NOP the inc esi and not the whole block for some voodoo reason:



```
loc_401178:
mov     eax, esi
nop
cmp     eax, 3
jz      short loc_40119B
```

And now it works!!!!

```
C:\Users\user\Desktop\hw3-samples2\samples>q1.exe test.txt 1 100
Hello
there
I'm
daniel
the
reverser!
```

```
C:\Users\user\Desktop\hw3-samples2\samples>
```

```
C:\Users\user\Desktop\hw3-samples2\samples>q1.exe ..\..\hw3-samples\samples\q1.exe 0 1000
MZÉ
=!qL=!This
program
cannot
be
run
in
DOS
mode.
$
Rich
```