Write a program that keeps reading filenames from the keyboard until the word "end" is introduced From all the filenames identify the file that contains the maximum number of capital letters and print the filename and it's content on the screen. The file can contain any number of chars. If there are more files with the same max of capital letters, then we chose the first. Program must have 2 modules with the function:

① one fct. that gets a filename as a parameter and computes the number of capital letter from that file

② one fct. that gets a filename as a parameter and print its name and content on the screen

```
bits 32
global start
extern exit, scanf
import exit msvcrt.dll
import scanf msvcrt.dll
extern capital_letters, print_file
```

```
Segment data use 32 class = data
file_name resb 25
max_cap_let_file_name resb 25
max_num_cap_let db 0
end_file db "end", 0
reading_format db "%s", 0

Segment code use 32 class = code
 Start:
      read_file_names:
          push file_name
          push reading_format
          call [scanf]
          add ESP, 4*2
          mov ESI, file_name
          mov EDI, end_file
          compare_file_name:
              cmp [ESI], byte 0
              je exit_compare_1
              Cmpsb
              jne exit_compare_2
              jmp compare_file_name
```

```
exit_compare_1:
cmp [EDi], byte 0
je  end_read_files_name
exit_compare_2:
push file_name
call capital_letters
cmp EAX, [max_cap_letters]
jbe  skip_update_max
     mov [max_cap_letters], EAX
     mov EDi, max_cap_file
copy_file_name:
     movsb
     cmp [ESi], byte 0
     je exit_copy_string
     jmp copy_file_name
exit_copy_string:
mov [EDi], byte 0 , skip_update_max:
jmp read_file_names
end_read_file_names:
push max_cap_letters_file
call print_file
```

```
        push dword 0
        call [exit]

bits 32
global capital_letters
extern fopen, fread, fclose
import fopen msvcrt.dll
import fread msvcrt.dll
import fclose msvcrt.dll

segment data use 32 class=data:
    read_mode db "r", 0
    file_disc dd -1
    buffer db 0
segment code use 32 class=code:
    capital_letters:
        mov ebx, 0
        push read_mode
        push dword [esp+4]
        call [fopen]
        pop esp, 8
        cmp eax, 0
        je end_module
```

```
mov [file_desc], eax

read_chars:                ; fread(buffer, size, count, desc)
    push file_desc
    push dword 1
    push buffer
    call [fread]
    add esp, 4*4
    cmp eax, 0
    je end_read
    cmp byte [buffer], 'A'
    jl no_increment
    cmp byte [buffer], 'Z'
    ja no_increment
    inc [counter]
no_increment:
    jmp read_chars
end_read:
    push dword [file_desc]
    call [fclose]
    add esp, 4
```

```
end_module:
mov rax, rbx
ret 4
```