

```
//Jon Ambrose & Simon Hartfield
//Reciever File

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#define CNTRLZ 0x9A
#define constkeysize 2

main(){
char Com_port_1[] ="COM1";
char Com_port_2[] ="rb";
char File_port_1[128] = "C:\\temp\\Recieved.txt"; //address of source file
char File_port_2[] ="wb";
char Key_port_1[128] = "C:\\temp\\Key.txt"; //address of source file
char Key_port_2[] ="rb";
char scans[] = "%s";

char keymessage[] = "\nPlease enter location of key file\n";
char dstmessage[] = "\nPlease enter location for decrypted file\n";
int data;
int count;
int keyext;

FILE * com;
FILE * dst;
FILE * key;

count = 0;

__asm
{
    lea eax, dstmessage //get source file address
    push eax
    call printf
    add esp, 4
    lea eax, File_port_1
    push eax
    lea eax, scans
    push eax
    call scanf
    add esp, 8

    lea eax, keymessage //get key file address
    push eax
    call printf
    add esp, 4
    lea eax, Key_port_1
    push eax
    lea eax, scans
    push eax
    call scanf
    add esp, 8

    lea eax, Com_port_2 //get pointer to COM Port
    push eax
    lea eax, Com_port_1
    push eax
    call fopen
    mov com,eax
    add esp, 8

    lea eax, File_port_2 //get pointer to source file
    push eax
    lea eax, File_port_1
    push eax
    call fopen
    mov dst,eax
    add esp, 8
```

```
lea eax, Key_port_2 //get pointer to key file
push eax
lea eax, Key_port_1
push eax
call fopen
mov key,eax
add esp, 8
```

//////////////////////////////////End of Initialisation//////////////////////////////////

nextchar:

```
mov eax, com //get next COM port character
push eax
call fgetc
mov data, eax
add esp, 4
cmp eax, CNTRLZ
jz end
```

encryption:

```
mov eax, key //get next key character
push eax
call fgetc
mov keyext, eax
add esp, 4
cmp eax, EOF
jz keyend

mov ebx, keyext
mov eax, data

xor eax, ebx //1st level of encryption, using external key
mov data, eax

mov eax, dst //send data using fputc
push eax
mov eax, data
push eax
call fputc
add esp, 8

mov eax, dst //flush the buffer to send character
push eax
call fflush
add esp, 4

jmp nextchar
```

keyend:

```
mov eax, key
push eax
call rewind
add esp, 4
jmp encryption
```

end:

```
mov eax, com //close com port
push eax
call fclose
add esp, 4

mov eax, dst //close dst file
push eax
call fclose
add esp, 4

xor eax,eax
```

```
xor ebx, ebx
```

```
}
```

```
return 0;
```

```
}
```