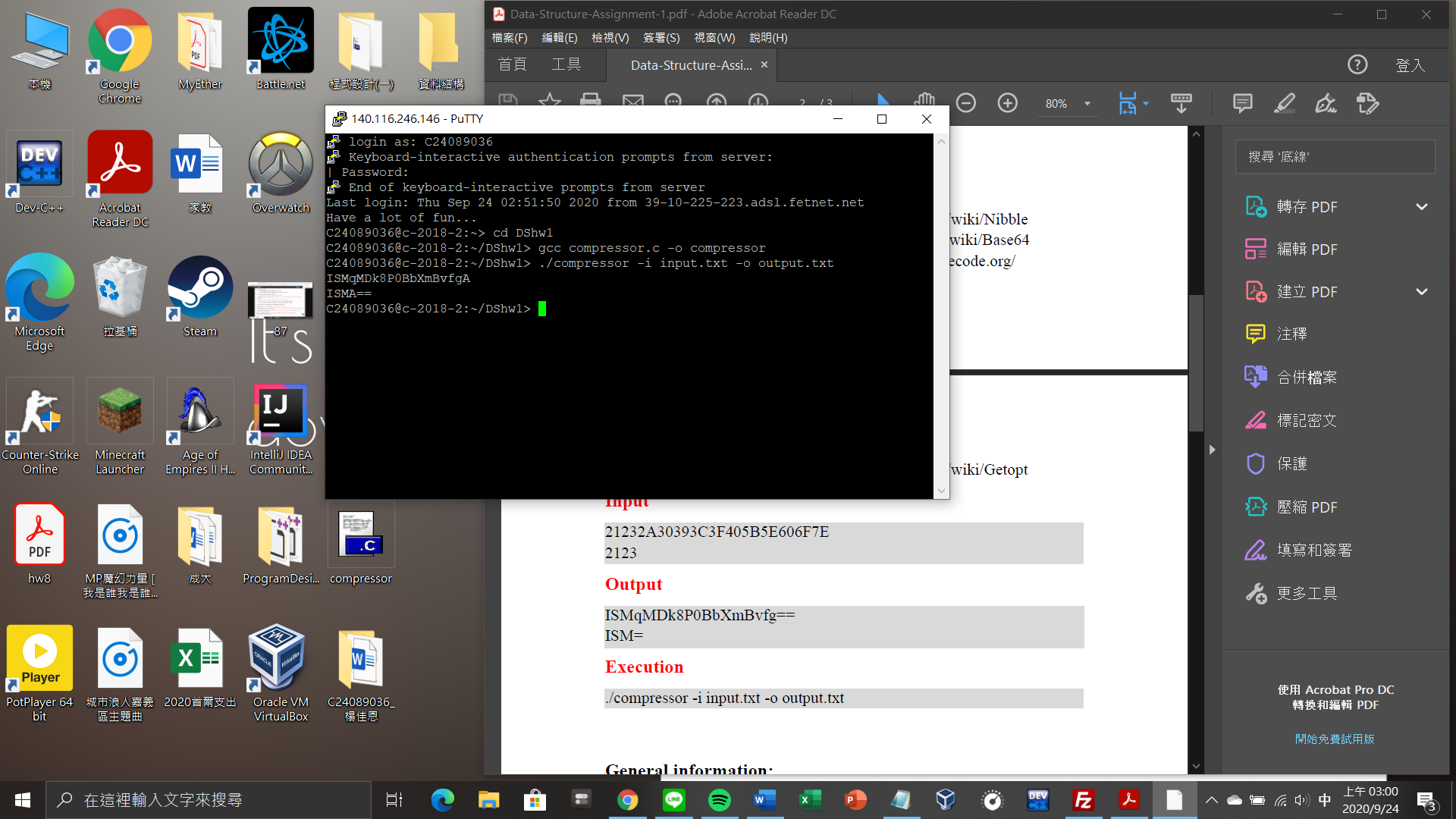
**Readme**

(1) result screenshot



(2) program architecture

(3) program functions

void substr(char \*dest, const char\* src, unsigned int start, unsigned int cnt)

Features

將資料分割成所需長度

Parameter

dest – 存放被分割的資料

src – 原始資料

start – 分割位置

cny – 分割長度

Return Value

None

int get\_num(const char\* data, unsigned int start)

Features

將字串轉成數字

Parameter

data – 帶轉換成數字的字串

start – 分割位置

Return Value

if data = 0, then return 0

if data = 1, then return 1

if data = 2, then return 2

if data = 3, then return 3

if data = 4, then return 4

if data = 5, then return 5

if data = 6, then return 6

if data = 7, then return 7

if data = 8, then return 8

if data = 9, then return 9

if data = A, then return 10

if data = B, then return 11

if data = C, then return 12

if data = D, then return 13

if data = E, then return 14

if data = F, then return 15

char double\_to\_bit (int a, int location)

Features

將十進位轉成二進位

Parameter

a – 待轉換的十進位數字

location – 指標位置

Return Value

二進位字串

int intBinary\_to\_decimal (int n)

Features

將二進位轉成十進位

Parameter

n - 待轉換的二進位數字

Return Value

十進位整數

int base\_num(const char\* src, unsigned int start)

Features

將bits資料分割成6bits一組並換算成整數

Parameter

src – 待處理的二進位字串

start – 分割位置

Return Value

十進位整數

void int\_to\_base(char target[32], int num[32], int word\_len, const char b64[], int word\_left)

Features

將處理完的十進位整數轉成BASE64字母

Parameter

target – BASE64字母存放處

num – 待處理的十進位整數

word\_len – 原始input.txt裡的資料長度

b64 – BASE64字母表

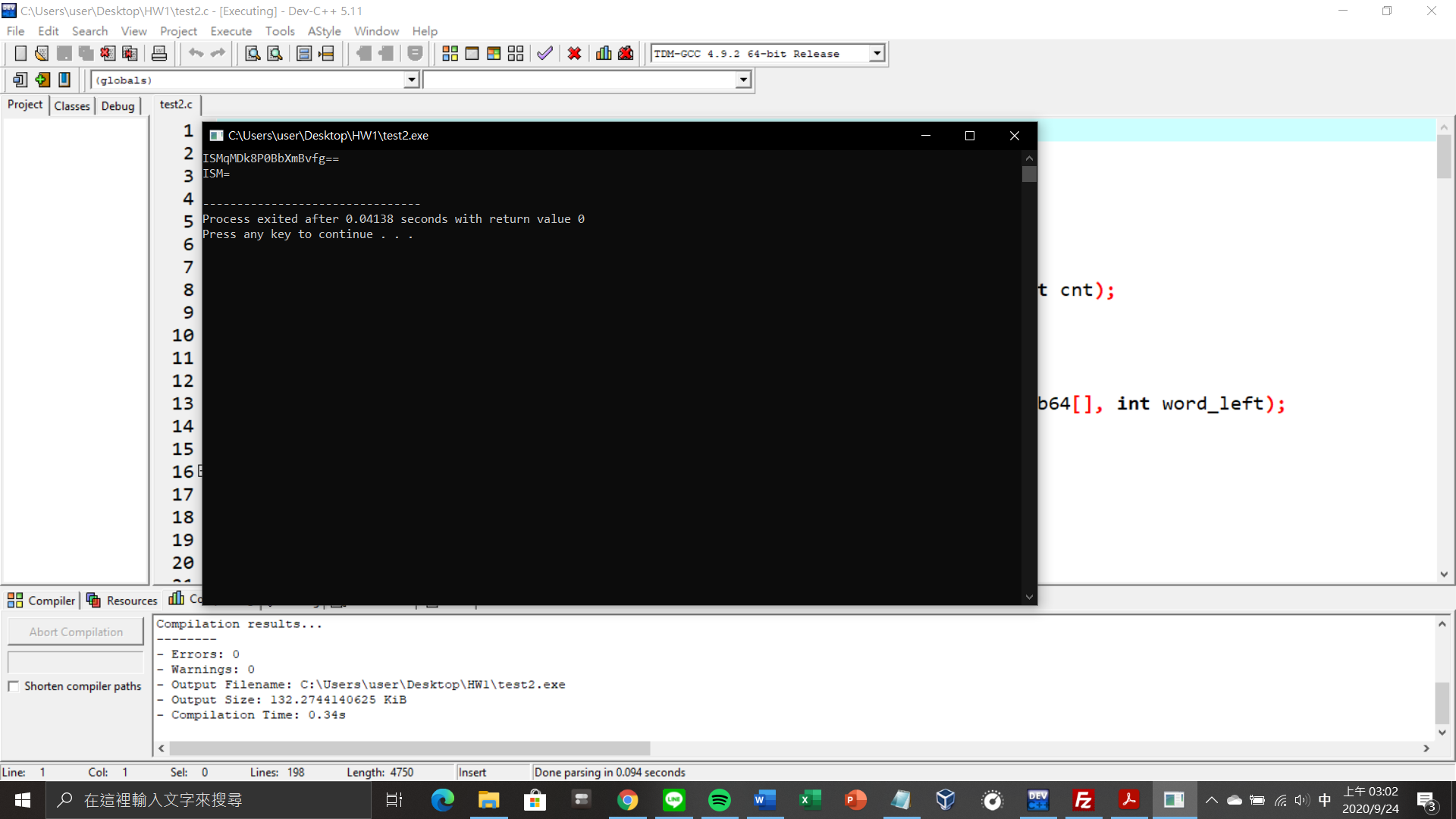
word\_left – 將二進位資料除以6後的餘數

Return Value

None

(4) how you design your program

作業程式設計的問題：在windows把檔名直接寫在程式中，輸出是正確的(如圖)。在linux透過commandline輸入檔名，輸出就變亂碼。



其相對應之解法：我不知道☹