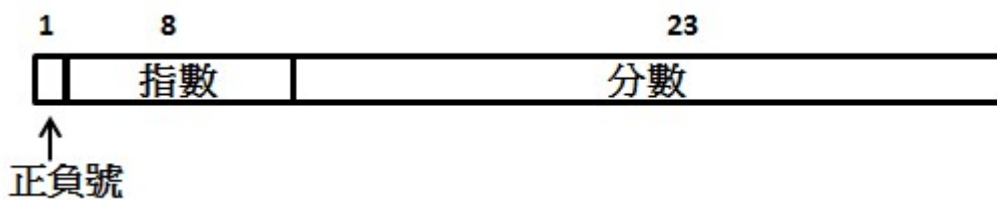


寫下列題目中的組合語言程式可以使用的指令有：MOV, ADD, SUB, ADC, SBB, NEG, INC, DEC, JMP, JB, JNB, JA, JNA, JG, JNG, JL, JNL, JC, JO, JP, JS, JNC, JNS, JNO, JNS, JNP, JZ, JNZ, JCXZ, LOOP, LOOPZ, LOOPNZ, CALL, RET, PUSH, POP, PUSHA, POPA, PUSHF, POPF, XCHG, XLAT, AND, OR, NOT, XOR, TEST, CMP, SHR, SHL, SAR, SAL, ROR, ROL, RCR, RCL, MUL, IMUL, DIV, IDIV, DAA, DAS, AAA, AAS, AAM, AAD, CBW, CWD, CDQ, STC, CLC, CMC, STD, CLD, STI, CLI, MOVSn, CMPSn, SCASn, LODSn, STOSn, REP, REPZ, REPNZ 或在課堂中講解過卻在這裡遺漏的其它指令。所有程式必須使用組合語言指令，不可使用 conditional directives。部分題目有建議的作法。只要達到題目的要求，同學可用其它方法。不過，太過繁雜的解題方法(例如，幾行指令就可解題，卻使用了三、四倍以上的指令)，雖可能得到正確結果，也無法達到滿分。

1. IEEE754 單精密度浮點數表示法格式如下所示：



其中，最高一個位元是正負號，0 代表正數，1 代表負數。接下來的 8 個位元是指數，而低階的 23 個位元是數值的分數部分。指數部分是以超 127 的方式存放的，即這個數值比實際的指數值大了 127。分數部分是將浮點數正規化後，數值的分數部分。浮點數正規化是將數值的絕對值化成 $1.XXX...2^N$ 的格式。其中，X 可能是 0 或 1。N+127 即是上列表示法的指數部分。XXX...即是分數部分。不足 23 位元的，在低階補以 0。例如，正整數+5 的浮點數表示是 $1.01*2^2$ 。正負號是 0，指數是 10000001_2 (129_{10})，分數是 $010...0$ 。請寫一程式片段，將存放在暫存器 EDX 的一個非零有正負號的整數轉成 IEEE754 單精密度浮點數，存放到暫存器 EAX。建議的轉換過程是

- (1) 檢查 EDX 的正負號，來設定 EAX 的正負號。若為負數，應求其絕對值，還是放在 EDX。設一計數 CNT 為 32。
- (2) 左移 EDX 一位元。計數 CNT 減一。
- (3) 檢查進位旗標。若是 0，回到步驟 2。
- (4) CNT 加 127，放到指數部分。
- (5) EDX 的高階 23 位元放到分數部分。

2. 下列是資料段的定義。

```
.DATA
TEXT1 BYTE 1000 DUP (?)
STR1 BYTE 100 DUP (?)
LEN1 WORD ?
LEN2 BYTE ?
```

現在，TEXT1 和 STR1 起的位置中各都已存有一個字串。對應的字串長度依序存放在 LEN1 和 LEN2 中。請寫一程式，在字串 TEXT1 中，找尋第一次字串 STR1 出現的位置。若找到，將對應的開始位址存放在暫存器 ESI 中，並將進位旗標清為 0。若找不到，應設進位旗標為 1，暫存器 ESI 的內容可以隨意。

3. 請寫一程式片段，將存放在資料段位置名稱 **SGNDEC** 的內容轉成有正負號的十進制字串，存放到由位置名稱 **DECOUT** 開始的位置中。有正負號的二進制數值轉成十進制字串的建議步驟是：
- (1) 檢查是正數或是負數。若是負數，則在字串的第一個字放個負號。若是正數，可放個正號，或不放。若是負數，計算其絕對值。
 - (2) 將數值除以 10，得到的餘數是個位數。
 - (3) 步驟 2 的商除以 10，得到的餘數是十位數。
 - (4) 重複前述的動作，依序得到百位數、千位數、...，直到商是 0。
 - (5) 將各位數數值轉成文字碼，存放到字串的適當位置。
 - (6) 放數碼 0 到字串後面。

4. 下列是資料段的定義，存有一個學生的各項成績 (期中考**MIDTERN**、期末考**FINAL**、四次作業**HW1-4**)和總成績**GRADE**。

.DATA

MIDTERM BYTE ?

FINAL BYTE ?

HW1 BYTE ?

HW2 BYTE ?

HW3 BYTE ?

HW4 BYTE ?

GRADE BYTE ?

請寫一程式片段，計算總成績。成績計算方式是期中和期末考各佔**40%**，作業平均佔**20%**。若有小數，將其四捨五入，得到整數的總成績。你的程式開始前，期中、期末考和各作業成績都已存入適當位置。建議的方法是

- (1) 將期中考和期末考分數相加後，乘以**40**。
- (2) 將作業成績相加後，乘以**5**。
- (3) 將前兩項結果相加後，再加**50**。
- (4) 將結果除以**100**。存到**GRADE**。

5. 請寫一個使用堆疊傳送參數的程序 **ARYSUM**，計算一個陣列中所有元素的和。呼叫 **ARYSUM** 來計算有 **NUM** 個元素的陣列 **ARY1** 的總和 **SUM**，執行的指令是

PUSH OFFSET ARY1

PUSH OFFSET NUM

PUSH OFFSET SUM

CALL ARYSUM

其中，**ARY1**、**NUM** 和 **SUM** 依序是存放陣列內容、元素個數和總和的位置，存放的都是 32 位元的資料。例如，資料段定義是

ARY1 SDWORD 1, -2, 3, 4, -5, 6, 7, 8, -9, 10

NUM DWORD 5

SUM SDWORD ?

則你的 **ARYSUM** 回轉(**RETURN**)後，位置 **SUM** 內將存有 1 (即 $1+(-2)+3+4+(-5)$)。程序回轉後，先前推入堆疊的資料，應已在程序中被彈出。