

問題 1. つぎの各プログラムを実行した後の、指定したレジスタの値を 10 進数または 16 進数 (数値の末尾に H を付けること) で答えよ. ただし, 各プログラムを実行する直前には, レジスタファイルの内容は下図の通りであるものとする.

20H	21H
21H	22H
22H	23H
23H	24H

(1)
 MOVF 20H,W
 MOVWF 21H
 MOVLW 22H
 MOVWF 23H
 21H の内容 = 2 1H, 23H の内容 = 2 2H

(2)
 MOVF 23H,W
 MOVWF 22H
 MOVF 21H,F
 MOVWF 20H
 20H の内容 = 2 4H, 22H の内容 = 2 4H

(3)
 MOVF 21H,W
 ADDWF 22H,F
 ADDWF 23H,F
 22H の内容 = 4 5H, 23H の内容 = 4 6H

(4)
 MOVF 21H,W
 ADDWF 22H,W
 ADDWF 23H,F
 22H の内容 = 2 3H, 23H の内容 = 6 9H

(5)
 DECF 20H,F
 INCF 21H,F
 20H の内容 = 2 0H, 21H の内容 = 2 3H

(6)
 LOOP:
 INCF 23H,F
 DECF 22H,F
 DECFSZ 20H,F
 GOTO LOOP
 22H の内容 = 0 2H, 23H の内容 = 4 5H

問題 2. 空欄 (下線部) を埋めてプログラムを完成せよ.
 (1) レジスタ 20H の内容を 12 倍する. ただし, 12 倍した結果は 255 以下であるものとする.

MOVF 20H,W ; 20H の内容を W へ退避する
 RLF 20H,F ; 20H を 2 倍する (元の 2 倍)
① 20H,F ; ??? (元の 3 倍)
② 20H,F ; ??? (元の 6 倍)
③ 20H,F ; ??? (元の 12 倍)

(2) 約 0.02 秒間だけ待ち合わせるソフトウェアタイマ. ただし, PIC の動作クロックは 20MHz とする.
 【ヒント】個々の命令の実行時間はクロックの 4 周期分である. ただし, GOTO 命令は 2 倍の 8 周期分かかる.

MOVLW 0FAH ; 繰り返しのために, 250 で,
 MOVWF 40H ; カウンタ A (40H) を初期設定
 LOOP_1:
 MOVLW ④ ; 繰り返しのために, ??? で,
 MOVWF ⑤ ; カウンタ B (??) を初期設定
 LOOP_2:
 NOP ; 何もしない命令 No Operation
 DECFSZ 30H,F ; カウンタ??を更新し,
 GOTO LOOP_2 ; 0 でなければ LOOP_2 から繰り返す
 DECFSZ ⑥ ; カウンタ??を更新し,
 GOTO LOOP_1 ; 0 でなければ LOOP_1 から繰り返す

(3) 30H 番地から始まるメモリブロック (ファイルレジスタ) に格納されている 10 個の整数データの中で, 値が 0 のデータの個数を 20H 番地のファイルレジスタに求める.

MOVLW ⑦ ; ブロックの先頭アドレスで,
 MOVWF FSR ; ポインタを初期設定
 MOVLW ⑧ ; 繰り返し制御用の
 MOVWF 21H ; カウンタを初期設定
 MOVLW 0 ; 値 0 のデータ数を数えるための
 MOVWF 20H ; カウンタを初期設定

LOOP:
⑨ INDF,F ; ブロックデータをチェックし,
 BTFSC ⑩ ; 0 ならば
 INCF 20H,F ; データ数カウンタを+1 する
 INCF FSR,F ; ポインタを更新し,
 DECFSZ 21H,F ; 繰り返し制御用カウンタを更新し,
 GOTO LOOP ; 0 でなければ,
 ; LOOP から繰り返す

【問題 2 の解答欄】

①	ADDWF	⑥	40H,F
②	RLF	⑦	30H
③	RLF	⑧	0AH
④	64H	⑨	MOVF
⑤	30H	⑩	STATUS,Z

⑨は、SUBWF も正解

問題3. つぎの(1)~(3)に示すのは、それぞれプログラムの一部分である。コメント文に記述した機能を持つように、空欄(下線部)を埋めてプログラム(の一部分)を完成せよ。

(1)

```
// 配列 a 内のデータの最大値を変数 max に求める。
// ただし、データの取り得る値の範囲は不明であり、
// 配列内には少なくとも 1 個のデータがあるものとする。
```

```
max = a[0];
for (int ①; ②; ③) {
    if (a[k] > max) {
        ④;
    }
}
```

(2)

```
// 配列 a 内の実数データを小さい順に並べ替える。
// ただし、データの取り得る値の範囲は不明であり、
// 配列内には 2 個以上のデータがあるものとする。
```

```
for (int j = 0; j < a.length - 1; j++) {
    double min = a[j];
    int min_j = j;
    for(int i = j + 1; i < a.length; i++) {
        if (min > a[i]) {
            min = ⑤;
            min_j = ⑥;
        }
    }
    a[min_j] = ⑦;
    a[j] = ⑧;
}
```

(3)

```
// 配列 x 内の整数データを大きい順に並べ替える。
// ただし、データの最大値は 100、最小値は 0 とし、
// 同じ値のデータはないものとする。
```

```
for (int i = 0; i <= 100 ; i++)
    bucket[i] = -1;
for (int i = 0; i < x.length; i++)
    bucket[⑨] = 1;
k = 0;
for (int i = 100, i >= 0 ; ⑩) {
    if (bucket[i] > 0) {
        x[k] = ⑪;
        ⑫;
    }
}
```

【問題3の解答欄】

①	k = 1	⑦	a[j]
②	k < a.length	⑧	min
③	k++	⑨	x[i]
④	max = a[k]	⑩	i--
⑤	a[i]	⑪	i
⑥	i	⑫	k++

①は、k = 0 でも正解