

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.

24H	24H
25H	23H
26H	22H
27H	21H

(1)

```

MOVF 24H,W
MOVWF 25H
MOVF 26H,W
MOVWF 27H
25H の内容 = _____, 27H の内容 = _____
    
```

(2)

```

MOVF 24H,W
MOVWF 25H
MOVF 26H,F
MOVWF 27H
25H の内容 = _____, 27H の内容 = _____
    
```

(3)

```

MOVLW 24H
MOVWF 25H
MOVLW 26H
MOVWF 27H
25H の内容 = _____, 27H の内容 = _____
    
```

(4)

```

MOVF 25H,W
ADDWF 26H,F
ADDWF 27H,F
26H の内容 = _____, 27H の内容 = _____
    
```

(5)

```

MOVF 25H,W
ADDWF 26H,W
ADDWF 27H,F
26H の内容 = _____, 27H の内容 = _____
    
```

(6)

```

DECF 26H,F
INCF 27H,F
26H の内容 = _____, 27H の内容 = _____
    
```

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 30H の内容を 9 倍する. ただし, 9 倍した結果は 255 以下であるものとする.

```

MOVF 30H,W ; 30H の内容を W へ退避する
BCF STATUS,C ; Carry フラグを 0 にする
RLF 30H,F ; 30H を 2 倍する (元の 2 倍)
① 30H,F ; ??? (元の 4 倍)
② 30H,F ; ??? (元の 8 倍)
③ 30H,F ; ??? (元の 9 倍)
    
```

(2) 約 0.05 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 ④ ; 繰り返しのために, ??? で,
MOVWF 20H ; カウンタ A (20H) を初期設定
LOOP_A:
MOVLW ⑤ ; 繰り返しのために, 200 で,
MOVWF 21H ; カウンタ B (21H) を初期設定
LOOP_B:
NOP ; 何もしない命令 No Operation
NOP ; 何もしない命令 No Operation
⑥ ; カウンタ B を更新し,
GOTO LOOP_B ; 0 でなければ LOOP_B から繰り返す
DECFSZ 20H,F ; カウンタ A を更新し,
GOTO LOOP_A ; 0 でなければ LOOP_A から繰り返す
    
```

(3) 20H 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 8 個の整数データの値を各々 1 だけ減らす.

```

MOVLW ⑦ ; ブロックの末尾アドレスで,
MOVWF FSR ; ポインタを初期設定
MOVLW ⑧ ; 繰り返し制御用の
MOVWF 28H ; カウンタを初期設定
    
```

```

LOOP:
⑨ INDF,F ; ブロックデータを 1 だけ減らし,
DECF FSR,F ; ポインタを更新し,
DECFSZ ⑩ ; 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		⑥	
②		⑦	
③		⑧	
④		⑨	
⑤		⑩	

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 x 内のデータの最小値を変数 min 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。
```

```
min = x[0];
for (int ①; ②; ③) {
    if (min > x[j]) {
        ④;
    }
}
```

(2)

```
// 配列 y 内の実数データを大きい順に並べ替える。
// 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 i = 0; i < y.length - 1; i++) {
    double max = y[i];
    int max_i = i;
    for(int j = i + 1; j < y.length; j++) {
        if (max < y[j]) {
            max = ⑤;
            max_i = ⑥;
        }
    }
    y[max_i] = ⑦;
    y[i] = ⑧;
}
```

(3)

```
// 配列 z 内の整数データを小さい順に並べ替える。
// 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 k = 0; k <= 100 ; k++)
    bucket[k] = 0;
for (int i = 0; i < z.length; i++)
    bucket[⑨] = 1;
i = 0;
for (int k = 0, k <= 100 ; ⑩) {
    if (bucket[k] > 0) {
        z[i] = ⑪;
        ⑫;
    }
}
```

【問題3の解答欄】

①		⑦	
②		⑧	
③		⑨	
④		⑩	
⑤		⑪	
⑥		⑫	