

You may NOT use a calculator. You may use only the provided reference materials. If a binary result is required, give the value in HEX. Assume all variables are in bank 0.

Part I: (70 pts)

- a. (10 pts) Write a PIC18 assembly code fragment to implement the following
(NOTE: these are **LONG** variables, which are 4 bytes!!!!)

```
long i, k;
```

```
i = i + k;
```

- b. (10 pts) Write a PIC18 assembly code fragment to implement the following:

```
signed int j, k;
```

```
do{  
    j = j << 1;  
}while(k >= j)
```

```
loop_top:  
    bcf     STATUS,C  
    rlc    j,f  
    rlc    j+1,f  
    movf   ____,w  
    ____  ____,w  
    movf   ____,w  
    ____  ____,w  
    b____ L1  
    b____ loop_top ;true loop top  
    bra   loop_exit ;exit  
L1  
    b____ loop_top ;true loop top  
loop_exit  
    ....rest of code....
```

- c. (5 pts) Write a PIC18 assembly code fragment to implement the following:
signed int k,j;

```
k = k >> 1 ;
```

- d. (10 pts) Implement the swap subroutine in PIC18 assembly language. Assume the parameters have been initialized by the calling function. Do NOT forget that this is a subroutine!!!!!!

```
// do swap
swap (unsigned char *s1, unsigned char *s2){
    unsigned char c;

    c = *s1;    // save *s1 value
    *s1 = *s2; // copy value
    *s2 = c;    // s2 gets old s1 value
}
```

```
;parameter space for swap subroutine
CBLOCK 0x020
s1:2, s2:2, c ; s1,s2 contains pointers to strings
ENDC
```

- e. (10 pts) Implement the following in PIC18 assembly. Assume the parameters have been initialized by the calling function. CAREFUL: *ptr* is a pointer to an INT value, and *val* is an INT value as well! The return value of '0' or '1' must be returned in the W register.

```
// do equality test
char test(unsigned int *ptr, unsigned int val){

    if (*ptr == val) return(0);
    else return(1);

}
```

```
;parameter space for test subroutine
CBLOCK 0x020
ptr:2, val:2
ENDC
```

- f. (10 pts) Implement the following in PIC18 assembly, which is a call to the subroutine 'test' of the previous problem. The assembly code should work regardless of where the parameter block for main is located. The '&i' passes the address of variable *i* to the test subroutine (this is the *ptr parameter). The value of *j* is passed to parameter 'val'. The return value of the subroutine *test* returns in W and should be written to 'k'.

```
main() {  
  int i,j;  
  char k;  
  k = test( &i, j);  
}
```

```
;allocation for main  
CBLOCK 0x????  
i:2, j:2, k  
ENDC  
  
;parameter space for test  
CBLOCK 0x020  
ptr:2, val:2  
ENDC
```

Assume the following memory contents at the START of EACH of these code fragments for problems g to h.

Location	Contents:
0x060	0xF2
0x061	0x50
0x062	0xA5
0x063	0x0B

W register = 0x02

g. (5 pts) Give the final contents of any changed REGISTERS or MEM locations.

```
lfsr    FSR0, 0x061
movff   PREINC0, 0x60
```

h. (5 pts) Give the final contents of any changed REGISTERS or MEM locations.

```
movlw   0x060
movwf   FSR0L
clrf    FSR0H
movff   0x063, INDF0
```

i. (5 pts) Give the final contents of any changed REGISTERS or MEM locations.

```
lfsr    FSR0, 0x061
movff   POSTDEC0, 0x60
```

Part II: (30 pts) Answer 10 out of the next 12 questions. Cross out the 2 questions that you do not want graded. Each question is worth 3 pts.

1. How does a call instruction differ from a goto instruction?
2. The value 0xE2 is a two's complement, 8-bit number. What is the decimal value?
3. Give the value of -3 as a 16-bit two's complement number.
4. Give the V, N flag settings after the operation $0x60 + 0x40$.
5. Give the V, N flag settings after the operation $0x80 - 0x1$.

6. In the code below, what is the value of i when the loop is exited?

```
signed char i;

i = 0x80;
while (i < -32) {
    i = i >> 1;
}
```

7. Define precisely what causes the stack *overflow* condition on the PIC18.
8. In the code below, j is a LONG variable which starts at memory location 0x020. Give the contents of locations 0x020,0x021,0x022,0x023 if the bytes are stored in LITTLE ENDIAN order:

```
long j;
j = 0xA433FA02;
```

9. In the C code fragment below, assume FSR0 is used to implement the 'ptr' variable. Write a PIC18 code fragment that implements the 'ptr++' operation. CAREFUL.....*ptr is a pointer to type 'int', which is two bytes long.

```
int *ptr;

ptr++;
```

10. Give the machine code for the 'bnn 0x0200' instruction below given the locations shown:

location		
0x0200	incf	0x020,f
0x0202	???	
0x0204	???	
0x0206	bnn	0x0200

11. Write a PIC18 assembly code fragment to implement the following:
signed int k,j;

k = k | j;

12. Of the signed comparisons below, circle the one that is TRUE!!!

unsigned char r, s;
signed char i, k;
signed int p, q;

r = 0x8F; s = 0x40;
i = 0x8F; k = 0x40;
p = 0xFF8F; q = 0x0040;

- a. i less than (<) k
- b. r less than (<) s
- c. p greater than (>) q