

Practice Exam 1

Name: Key

You will have 60 minutes for this exam. This exam is open book and note. Please allow some time to check your work. If you need extra space, write on the back. There are a total of 30 points on this exam.

1. (4 points) What is the output of the following C++ program, assuming that characters are encoded in ASCII as usual? Recall that $\&$ is the bitwise-AND operator and $|$ is bitwise-OR.

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    char msgA[13] = "Hello World!"; // 13 characters, including terminating NUL
    char msgB[13];
```

```
    for (int i = 0; i < 13; i++) {
        char ch = msgA[i];
```

```
        if ((ch & 96) == 64) {
            msgB[i] = ch | 32;
```

```
        } else {
            msgB[i] = ch;
```

```
        }
```

```
    cout << msgA << endl;
    cout << msgB << endl;
```

```
}
```

Hello World!
hello world!

\leftarrow true if in range 64-95
i.e., upper-case
 \leftarrow switch to lower-case
 \leftarrow unchanged

2. (6 points) Give the result of evaluating each of the following expressions, and tell which of the Pep/8 status flags (NZVC) would be set afterward. All numbers are in hexadecimal, as should be your results.

(a) $7531 + 3FFF$

$N=1$	$Z=0$	$V=1$	$C=0$	$\begin{array}{r} 7531 \\ + 3FFF \\ \hline B530 \end{array}$	$\begin{array}{r} 0111\ 0101\ 0011\ 0001 \\ + 0011\ 1111\ 1111\ 1111 \\ \hline 1011\ 0101\ 0011\ 0000 \end{array}$
-------	-------	-------	-------	--	--

(b) $FFFE + 8001$

$N=0$	$Z=0$	$V=1$	$C=1$	$\begin{array}{r} FFFE \\ + 8001 \\ \hline 7FFF \end{array}$	$\begin{array}{r} 1111\ 1111\ 1111\ 1110 \\ + 1000\ 0000\ 0000\ 0001 \\ \hline 0111\ 1111\ 1111\ 1111 \end{array}$
-------	-------	-------	-------	--	--

3. (8 points) Consider the byte 0101 0001:

(a) What is its hexadecimal representation?

51

(b) What does it represent when interpreted as an unsigned integer?

81_{10}

(c) What does it represent when interpreted as a two's complement integer?

$+81_{10}$

(d) What does it represent when interpreted as an excess-127 integer?

-46_{10}

(e) What does it represent when interpreted as an ASCII character?

Q

(f) What does it represent when interpreted as a Pep/8 machine instruction? (Give the assembly mnemonic and addressing mode)

CHARO direct

(g) What does it represent when interpreted as a floating-point number (with a sign bit, a three-bit exponent, and a four-bit significand)?

$$+1.0001 \times 2^2 = 100.01 = 4.25_{10}$$

(h) What does it represent when interpreted as a fixed-point number (with the binary point in the middle)?

$$0101.0001 = 5.0625_{10}$$

(or, $5\frac{1}{16}$)

4. (6 points) Convert the following C++ program to Pep/8 Machine Language (you may assume that the result of $1 + 1$ is a single digit, but you should not simply hard-code the character '2'...):

```
#include <iostream>
using namespace std;

int main() {
    cout << '4';
    cout << 1 + 1;
    cout << endl;
}
```

0000:	50 00 34	; character output: '4', immed.
0003:	C0 00 01	; load A: 1, immed.
0006:	70 00 01	; add A: 1, immed.
0009:	A0 00 30	; or A: 0x30, immed.
		(convert 0-9 to ASCII digit)
000C:	F1 00 16	; store byte A: 0016, direct
000F:	51 00 16	; character output: 0016, direct
0012:	50 00 0A	; character output: '\n', immed.
0015:	00	; stop
0016:	xx	; temporary byte storage

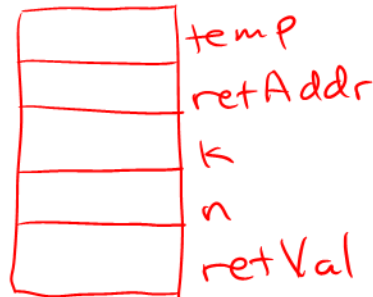
5. (6 points) Consider the following recursive function in C++ (it computes entries in a table known as the Bell Triangle, which can be used to count the number of ways to partition n distinct objects into nonempty sets):

```
int B(int n, int k)
{
    int temp;

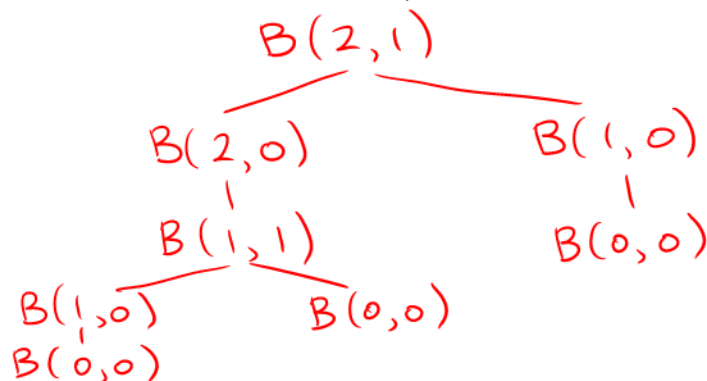
    if (k == 0 && n == 0) return 1;
    if (k == 0) return B(n-1, n-1);

    temp = B(n, k-1);
    return temp + B(n-1, k-1);
}
```

- (a) Sketch the layout of items in a stack frame for this function:



- (b) Show the call tree for the function call $B(2, 1)$:



- (c) If each `int` occupies 2 bytes, and memory addresses also take 2 bytes, then what is the minimum number of bytes needed on the runtime stack to process the function call $B(2, 1)$?

$$5 \text{ stack frames} \times 10 \text{ bytes/frame} \\ = 50 \text{ bytes}$$