1.) Consider the primitive C types **short** and **char** on our lab's x86_64 systems. The type **short** is used to represent 16-bit two's complement numbers, and the type **char** is used to represent 8-bit two's complement numbers. What output is printed after executing the following C code snippet?

```c
short s = 0x12b4;
char c = (char) s;
printf("%d", c);
```

*Hint:* The `printf` format string character `%d` is used to print the decimal value of an integer type, including the primitive type `char`.

-76

2.) What is the binary representation of the decimal number 29.3125?

11101.0101

3.) For this question, please consider the following **32-bit floating point** representation of a number `x`:

\[
x = \begin{array}{c}
1 & 100 & 0001 & 1010 & 1101 & 0000 & 0000 & 0000 & 0000 \\
\end{array}
\]

Recall that for a **normalized** 32-bit floating point number, the `exp` field has 8 bits. Therefore, the bias is \(2^8 - 1 = 127\).

What is the decimal value of `x`?

\[
\begin{align*}
Sign &= 1 \\
Exp &= 10000011 = 131 - 127 = 4 \\
M &= 1.0101101 \\
-21.625
\end{align*}
\]
4.) Consider the following C code:

```c
// Returns the result of x-y. If positive overflow occurs, prints an error message // and exits.
int subtract(int x, int y)
{
    int result = x - y;

    if( condition ){
        printf("Positive overflow occurred\n");
        exit(1);
    }
    else{
        return result;
    }
}
```

Provide C code that would replace `condition` in the code in order to detect when positive overflow occurs.

\[ x > 0 \land y < 0 \land \text{result} < 0 \]

5.) Consider the following C code snippet:

```c
int arr[4];
int *ptr = &arr[2];
arr[0] = 7;
*(ptr+1) = 30;
*(arr+2) = 70;
int val = arr[2];
*ptr = 3;
```

Assume that `ints` are stored in 4 bytes and pointers are stored in 8 bytes. The memory addresses (in decimal) associated with the first byte of each of the variables is as follows:

- `arr` 100
- `ptr` 116
- `val` 124

Indicate what values would be stored at the end of the code:

- `arr[0] = 7`
- `arr[1] = ???`
- `arr[2] = 3`
- `arr[3] = 30`
- `ptr = 108`
- `val = 70`