

Purpose: Investigate how numbers are stored, precision, casting, a look at some operators.

1a) Type in the following C program which prints the size of a “char” type, but duplicate the `printf()` line multiple times, changing it so it also prints the size of types: `unsigned char`, `short`, `unsigned short`, `int`, `unsigned int`, `long`, `unsigned long`, `long long`, `unsigned long long`, `float`, `double`, `char*`, and `long*`. Compile, run and record the results.

```
#include <stdio.h>
int main() {
    printf("sizeof char = %d\n", sizeof(char));
    return 0;
}
```

1b) Get the short program below from the web site using the following from the command line:

```
$ wget http://web.eece.maine.edu/eason/ece177/lab3b.c
```

Compile and run.

Repeat the program but set “x” to -78. Be prepared to explain every single result to a TA.

```
#include <stdio.h>

int main() {
    int x = 78;
    float f = 3.14159256;
    double db = 2.718281828;

    printf("x = %d*%5d*%05d*%c\n", x, x, x, x);
    printf("x = %u*%5u*%05u*%c\n", x, x, x, x);
    printf("x = %x*%5x*%05x*%c\n", x, x, x, x);
    printf("x = %X*%5X*%05X*%c\n", x, x, x, x);

    printf("f = %f*%10f*%010f\n", f, f, f);
    printf("db = %lf*%10lf*%010lf\n", db, db, db);
    printf("f = %f*%5.2f*%05.2f\n", f, f, f);
    printf("db = %f*%5.2f*%05.2f\n", db, db, db);

    printf("Hello, world!\n");
    printf("%s", "Hello, world!\n");
}
```

Note that the code for the following exercises may give compiler warnings. Normally you will want to fix your code so you don't get any warnings, as it usually indicates undesirable behavior (or at minimum it reveals tricky things that make your code harder to understand and verify).

2a) Write a short program which includes multiple lines similar to the following. The other lines should substitute the numbers 150, -150, 300, -1, and 65535. Run and record the results. Also give each result in 16-bit binary.

```
printf("%d\t%u\n", (short)45, (unsigned short)45);
```

2b) Also, include multiple pairs of lines similar to the following, where `c` is declared as a `char` type. The other pairs of lines should substitute the values of 150, -150, 300, -1, and 65535. Run and record the results. Also give each result in 16-bit binary.

```
c = 45;
printf("%d\t%d\t%u\n", 45, (short)c, (unsigned short)c);
```

2c) Also, include multiple pairs of lines similar to the following, where `uc` is declared as an `unsigned char` type. The other pairs of lines should substitute the values of 150, -150, 300, -1, and 65535. Run and record the results. Also give each result in 16-bit binary.

```
uc = 45;
printf("%d\t%d\t%u\n", 45, (short)uc, (unsigned short)uc);
```

3) Write a program to compute and print the date of Easter using the algorithm below and integer arithmetic. Run the program seven times, setting the year to 2018, 2019, 2020, 2021, 2022, 2023, and 2024. The result should be printed as "Date of Easter in 2018 is 4/1".

ALGORITHM TO DETERMINE THE DATE OF EASTER:

Divide	by	Quotient	Remainder
Year	19	-	A
Year	100	B	C
B	4	D	E
B + 8	25	F	-
B - F + 1	3	G	-
19*A + B - D - G + 15	30	-	H
C	4	I	K
32 + 2*E + 2*I - H - K	7	-	L
A + 11*H + 22*L	451	M	-
H + L - 7*M + 114	31	N	P

Month = N (March = 3, April = 4)

Day = P + 1