Code Examples from Lectures on 5-2_NumericTypes

`printVal.c` (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/5-2_NumericTypes/code/printVal.c)

A simple program demonstrating the use of printf for ints

And demonstrating that C can understand decimal and hex formats and display the same number in different ways with different conversion specifiers (ie %d for decimal and %x for hexadecimal in this example)

```
#include <stdio.h>

int main(void) {

    //print out a decimal number as a decimal then in hexadecimal
    printf("%d \n",36769);
    printf("%x \n",36769);

    //print out a hexadecimal number as a decimal then in hexadecimal
    printf("%d \n",0x8fa1 );
    printf("%x \n",0x8fa1);

    return 0;
}
```

`overflow.c` (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/5-2_NumericTypes/code/overflow.c)

A simple program demonstrating int overflow

Compile and run with dcc as usual. It detects our overflow and stops the program.

To see what would have happened compile with gcc -Wall -Werror -O -o overflow overflow.c and then run ./overflow

```
#include <stdio.h>


int main(void) {

    int big, bigPlus1, bigTimes2, reallyBig;

    big = 2147483647;
    bigPlus1 = big + 1;
    bigTimes2 = big * 2;
    reallyBig = bigPlus1 * 2;
    printf("big=%d bigPlus1=%d\n", big, bigPlus1);
    printf("bigTimes2=%d ", bigTimes2);
    printf("reallyBig=%d\n", reallyBig);

    return 0;
}
```

`overflowFix1.c` (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/5-2_NumericTypes/code/overflowFix1.c)

A simple program demonstrating overflow

```
#include <stdio.h>

//We tried to fix overflow.c it by using long long instead of int
//It helped with the overflow in the original program but long long is still not
//big enough for our new 'reallyReallyBig' variable


int main(void) {

    long long big, bigPlus1, bigTimes2, reallyBig;
    long long reallyReallyBig;
    big = 2147483647;
    bigPlus1 = big + 1;
    bigTimes2 = big * 2;
    reallyBig = bigPlus1 * 2;
    reallyReallyBig = bigPlus1*bigPlus1*bigPlus1;
    printf("big=%lld bigPlus1=%lld\n", big, bigPlus1);
    printf("bigTimes2=%lld ", bigTimes2);
    printf("reallyBig=%lld\n", reallyBig);
    printf("reallyReallyBig=%lld\n", reallyReallyBig);

    return 0;
}
```

overflowFix2.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/5-2_NumericTypes/code/overflowFix2.c)

A simple program demonstrating overflow

```
#include <stdio.h>

//We tried to fix overflowFix.c it by using long long instead of int
//It helped with the overflow in the original program but long long is still not
//big enough for our new 'reallyReallyBig' variable. So now we are trying unsigned long long. //It is stil


int main(void) {

    unsigned long long big, bigPlus1, bigTimes2, reallyBig;
    unsigned long long reallyReallyBig;
    big = 2147483647;
    bigPlus1 = big + 1;
    bigTimes2 = big * 2;
    reallyBig = bigPlus1 * 2;
    reallyReallyBig = bigPlus1*bigPlus1*bigPlus1;
    printf("big=%llu bigPlus1=%llu\n", big, bigPlus1);
    printf("bigTimes2=%llu ", bigTimes2);
    printf("reallyBig=%llu\n", reallyBig);
    printf("reallyReallyBig=%llu\n", reallyReallyBig);

    return 0;
}
```

overflowFix3.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/5-2_NumericTypes/code/overflowFix3.c)

A simple program demonstrating overflow

```
#include <stdio.h>

//double's can represent all the variables here. But eventually double variables will reach a
//limit too. See overflowDouble.c


int main(void) {

    double big, bigPlus1, bigTimes2, reallyBig;
    double reallyReallyBig;
    big = 2147483647;
    bigPlus1 = big + 1;
    bigTimes2 = big * 2;
    reallyBig = bigPlus1 * 2;
    reallyReallyBig = bigPlus1*bigPlus1*bigPlus1;
    printf("big=%lf bigPlus1=%lf\n", big, bigPlus1);
    printf("bigTimes2=%lf ", bigTimes2);
    printf("reallyBig=%lf\n", reallyBig);
    printf("reallyReallyBig=%lf\n", reallyReallyBig);

    return 0;
}
```

overflowDouble.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/5-2_NumericTypes/code/overflowDouble.c)

A simple program demonstrating overflow try with gcc and then dcc

```
#include <stdio.h>
#include <math.h>

//Even double will reach its limits when the numbers get bigger
//What happens in this example?

int main(void) {

    double big, bigp1, bigt2, reallyBig;

    big = 2147483647;

    bigp1 = big + 1;
    bigt2 = big * 2;
    reallyBig = bigp1 * 2;
    printf("big=%.20lf bigp1=%.20lf\n", big, bigp1);
    printf("bigt2=%.20lf ", bigt2);
    printf("reallyBig=%.20lf\n", reallyBig);
    double reallyReallyBig = pow(reallyBig,31);
    double reallyReallyBigger = pow(reallyBig,32);
    printf("%lf\n",reallyReallyBig);
    printf("%lf\n",reallyReallyBigger);

    return 0;
}
```

limits.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/5-2_NumericTypes/code/limits.c)

A program to demonstrate the constants available in limits.h to find the max and min values for integer types and how to find the size in bytes of different types

```
#include <stdio.h>
#include <limits.h>

// more /usr/include/limits.h

int  main(void){

    printf("ints are stored in %d bytes\n",sizeof (int));
    printf("long longs are stored in %d bytes\n",sizeof (long long));

    printf("The biggest integer is            %d\n",INT_MAX);
    printf("The smallest integer is           %d\n",INT_MIN);
    printf("The biggest long long is          %lld\n",LLONG_MAX);
    printf("The biggest unsigned long long is %llu\n",ULLONG_MAX);

    return 0;
}
```

precision.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/5-2_NumericTypes/code/precision.c)

Demonstrate approximate representation of reals

For example value 0.1 can not be precisely represented as a real

```
#include <stdio.h>

int main(void) {
    double a, b;
    a = 0.1;
    b = 1 - (a + a + a + a + a + a + a + a + a + a);
    printf("%.20lf\n", b);

    double x = 0.1;
    printf("%.20lf\n",x);

    double y = 0.00000005;
    printf("%.20lf\n",y);
    double z = 999999999999999.00000005;
    printf("%.20lf\n",z);

    return 0;
}
```