Christopher Martin, Jennifer Wolfe, Geoffrey Corey
April 19, 2013
CS 472 Lab 1

```c
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <math.h>
#include <endian.h>
#include <stdlib.h>
#include <float.h>

union doubleBits {
        double dbl;
        struct {
#if _BYTE_ORDER == _LITTLE_ENDIAN
                unsigned int mantissaLow:32;
                unsigned int mantissaHigh:20;
                unsigned int exponent:11;
                unsigned int sign:1;
#else
                unsigned int sign:1;
                unsigned int exponent:11;
                unsigned int mantissaHigh:20;
                unsigned int mantissaLow:32;
#endif
        } components;
};

struct dblstring {
        union convertDoubleString {
                double dbl;
                char string[8];
        }doublestring;
        char end;
};

double frexp(double x, int *exp);

int main(int argc, char **argv)
{

        double number;
        double mantissa;
        int exponent;
        struct dblstring charstr;

        printf("Enter a number:\n");
        scanf("%lf", &number);

        mantissa = frexp(number, &exponent);

        /* Double, what is mantissa? */
        printf("Treated as a double, mantissa =\t %lf\n", mantissa);
```

```
        /* Double, what is sign */
        printf("Treated as a double, sign =\t  %s\n", (number < 0) ? "negative" : "positive");

        /* Double, what is exponent? */
        printf("Treated as a double, exponent =\t %d\n", exponent);

        /* Long, value? */
        printf("Treated as a long, the value =\t %ld\n", (long)number);

        /* Long, sign? */
        printf("Treated as a long, sign =\t %s\n", ((long)number < 0) ? "negative" : "positive");

        /* 8 chars */
        charstr.doublestring.dbl = number;
        charstr.end = '\0';
        printf("Treated as 8 characters =\t %s\n", charstr.doublestring.string);

        exit(EXIT_SUCCESS);
}

double frexp(double x, int *exp)
{

        union doubleBits number;

        number.dbl = x;

        switch(number.components.exponent){
        case 0:
                /* 0 or subnormal value */
                if(!(number.components.mantissaLow | number.components.mantissaHigh)){
                        *exp = 0;
                } else {
                        number.dbl *= 0x1.0p514; /* 2^64, restores the number. */
                        *exp = number.components.exponent - 1536;
                        number.components.exponent = 1022;
                }
                break;

        case 2047:
                /* Infinity or NaN, *exp value is unspecified */
                break;

        default:
                /* normal floating point number */
                *exp = number.components.exponent - 1022;
                number.components.exponent = 1022;
                break;
        }

        return (number.dbl);
}
```

```
/*
 * Citing resources:
 *
 * frexp: http://src.gnu-darwin.org/src/lib/libc/gen/frexp.c
 * unions + endianess: http://www.opensource.apple.com/source/Libc/Libc-825.25/fbsdcompat/fpmath.h
 *
 */
```