May 16, 04 19:32 **eg53−bitops.c** Page 1/1

```c
/*
 * Illustrate bit AND, OR and SHIFT operations.
 *
 * This example read an MPEG−1 video header and retrieve
 * information about width and height of the video.
 */

#include <stdio.h>
#include <inttypes.h>

int main(int argc, char *argv[])
{
    short w, h;
    if (argc < 2) {
        fprintf(stderr, "usage: %s <.MPG file>\n", argv[0]);
        return 1;
    }

    FILE *f = fopen(argv[1], "rb");
    if (f == NULL) {
        fprintf(stderr, "cannot open %s for reading.\n", argv[1]);
        return 1;
    }

    // MPEG header consists of a 4 bytes "magic code",
    // follows by 12 bits for width and 12 bits for
    // height.

    unsigned char c[4];

    fread(c, 4, 1, f);
    if (c[0] != 0 || c[1] != 0 || c[2] != 1 || c[3] != 0xb3)
    {
        fprintf(stderr, "%s is not a valid MPEG file\n", argv[1]);
        return 2;
    }
    fread(c, 4, 1, f);

    w = (c[0] << 4) | ((c[1] & 0xF0) >> 4);
    h = ((c[1] & 0x0F) << 8) | c[2];

    printf("%d %d\n", w, h);

    fclose(f);
}
```

May 16, 04 19:37 **eg54−bits.c** Page

```c
/*
 * Another common use of bitwise | operation is to compactly
 * indicate options.
 */
#include <sys/types.h>
#include <sys/stat.h>

int main()
{
    chmod("eg54−bits.c", S_IRUSR | S_IWUSR | S_IXUSR);
}
```

```
/*
 * Besides struct, two other user−define data types
 * is enum (short for enumeration) and union.
 *
 * enum provides an alternate way to defining integer constants.
 * union provides a convinient way to interpret content of a memory
 *   location as different type.
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef enum {
    TYPE_INT,
    TYPE_FLOAT,
    TYPE_STRING,
} argument_type;

typedef struct argument {
    argument_type t;
    union {
        int x;
        float f;
        char *s;
    } value;
} argument;

int is_integer(char *s);
int is_float(char *s);

int main(int argc, char *argv[])
{
    int i;
    argument *a = (argument *)malloc(sizeof(argument)*argc);

    printf("sizeof argument_type is %d\n", sizeof(argument_type));
    printf("sizeof argument is %d\n", sizeof(argument));

    for (i = 0; i < argc; i++)
    {
        if (is_integer(argv[i]))
        {
            a[i].t = TYPE_INT;
            a[i].value.x = atoi(argv[i]);
        }
        else if (is_float(argv[i]))
        {
            a[i].t = TYPE_FLOAT;
            a[i].value.x = atof(argv[i]);
        }
        else
        {
```



```
            a[i].t = TYPE_STRING;
            a[i].value.s = argv[i];
        }
    }
    return 0;
}

/*
 * Two functions to check for simple integer and float.
 * They are _not_ excellent examples of readable code!
 */
int is_integer(char *s)
{
    char *c = s−1;
    while (*(++c))
        if (*c < '0' || *c > '9')
            return 0;
    return 1;
}

int is_float(char *s)
{
    int dot_count = 0;
    char *c = s−1;
    while (*(++c)) {
        if (*c == '.') {
            if (dot_count)
                return 0;
            dot_count = 1;
        } else if (*c < '0' || *c > '9')
            return 0;
    }
    return 1;
}
```

May 16, 04 20:13 **eg56−register.c** Page 1/1

```c
/*
 * Introducing the "register" type modifier, which
 * give hints to compiler that the integer will be
 * access frequently, and so it is better to keep
 * them in register.
 */
int main()
{
    register int i, j;

    char matrix[100][100];
    for (i = 0; i < 100; i++)
        for (j = 0; j < 100; j++)
            matrix[i][j] = 0;
}
```

May 16, 04 20:19 **eg57−extern.c** Page

```c
/*
 * Introducing "extern" keyword, which indicates that a
 * variable is defined in another module.
 */
#include <stdio.h>

extern float sqrt_table[];

int main(int argc, char *argv[])
{
    if (argc < 2) {
        fprintf(stderr, "usage: %s <number>\n", argv[0]);
        fprintf(stderr, "<number> must be between 0 and 255\n");
        exit(1);
    }
    char *remaining;
    int result =  strtol(argv[1], &remaining, 10);

    if (result >= 0 && result <= 255) {
        printf("%f\n", sqrt_table[result]);
        return 0;
    } else {
        fprintf(stderr, "usage: %s <number>\n", argv[0]);
        fprintf(stderr, "<number> must be between 0 and 255\n");
        return 2;
    }
}
```

```
/*
 * An example of a global variable declared in one file and
 * used as extern in another (eg57−extern.c)
 */
float sqrt_table[] = {
    0.0, 1.0, 1.41421356237, 1.73205080757, 2.0, 2.2360679775,
    2.44948974278, 2.64575131106, 2.82842712475, 3.0, 3.16227766017,
    3.31662479036, 3.46410161514, 3.60555127546, 3.74165738677,
    3.87298334621, 4.0, 4.12310562562, 4.24264068712, 4.35889894354,
    4.472135955, 4.58257569496, 4.69041575982, 4.79583152331,
    4.89897948557, 5.0, 5.09901951359, 5.19615242271, 5.29150262213,
    5.38516480713, 5.47722557505, 5.56776436283, 5.65685424949,
    5.74456264654, 5.83095189485, 5.9160797831, 6.0, 6.0827625303,
    6.16441400297, 6.2449979984, 6.32455532034, 6.40312423743,
    6.48074069841, 6.5574385243, 6.63324958071, 6.7082039325,
    6.78232998313, 6.8556546004, 6.92820323028, 7.0, 7.07106781187,
    7.14142842854, 7.21110255093, 7.28010988928, 7.34846922835,
    7.4161984871, 7.48331477355, 7.54983443527, 7.61577310586,
    7.68114574787, 7.74596669241, 7.81024967591, 7.87400787401,
    7.93725393319, 8.0, 8.0622577483, 8.12403840464, 8.18535277187,
    8.24621125124, 8.30662386292, 8.36660026534, 8.42614977318,
    8.48528137424, 8.54400374532, 8.60232526704, 8.66025403784,
    8.71779788708, 8.77496438739, 8.83176086633, 8.88819441732,
    8.94427191, 9.0, 9.05538513814, 9.11043357914, 9.16515138991,
    9.21954445729, 9.2736184955, 9.32737905309, 9.38083151965,
    9.43398113206, 9.48683298051, 9.53939201417, 9.59166304663,
    9.64365076099, 9.69535971483, 9.74679434481, 9.79795897113,
    9.8488578018, 9.89949493661, 9.94987437107, 10.0, 10.0498756211,
    10.0995049384, 10.1488915651, 10.1980390272, 10.246950766,
    10.295630141, 10.3440804328, 10.3923048454, 10.4403065089,
    10.4880884817, 10.5356537529, 10.5830052443, 10.6301458127,
    10.677078252, 10.7238052948, 10.7703296143, 10.8166538264,
    10.8627804912, 10.9087121146, 10.9544511501, 11.0, 11.0453610172
,
    11.0905365064, 11.1355287257, 11.1803398875, 11.2249721603,
    11.2694276696, 11.313708499, 11.3578166916, 11.401754251,
    11.4455231423, 11.4891252931, 11.5325625947, 11.5758369028,
    11.6189500386, 11.6619037897, 11.7046999107, 11.7473401245,
    11.7898261226, 11.8321595662, 11.874342087, 11.9163752878,
    11.9582607431, 12.0, 12.0415945788, 12.0830459736, 12.124355653,
    12.1655250606, 12.2065556157, 12.2474487139, 12.2882057274,
    12.3288280059, 12.3693168769, 12.409673646, 12.449899598,
    12.4899959968, 12.5299640861, 12.56980509, 12.6095202129,
    12.6491106407, 12.6885775404, 12.7279220614, 12.7671453348,
    12.8062484749, 12.8452325787, 12.8840987267, 12.9228479833,
    12.9614813968, 13.0, 13.0384048104, 13.0766968306, 13.1148770486
,
    13.152946438, 13.1909059583, 13.2287565553, 13.2664991614,
    13.3041346957, 13.3416640641, 13.3790881603, 13.416407865,
    13.4536240471, 13.4907375632, 13.5277492585, 13.5646599663,
    13.6014705087, 13.638181697, 13.6747943312, 13.7113092008,
    13.7477270849, 13.7840487521, 13.8202749611, 13.8564064606,
    13.8924439894, 13.9283882772, 13.9642400438, 14.0, 14.0356688476
```



```
,
    14.0712472795, 14.1067359797, 14.1421356237, 14.1774468788,
    14.2126704036, 14.2478068488, 14.2828568571, 14.3178210633,
    14.3527000944, 14.3874945699, 14.4222051019, 14.4568322948,
    14.4913767462, 14.5258390463, 14.5602197786, 14.5945195193,
    14.6287388383, 14.6628782986, 14.6969384567, 14.7309198627,
    14.7648230602, 14.7986485869, 14.8323969742, 14.8660687473,
    14.8996644258, 14.9331845231, 14.9666295471, 15.0, 15.033296
,
    15.0665191733, 15.0996688705, 15.1327459504, 15.1657508881,
    15.1986841536, 15.2315462117, 15.2643375225, 15.2970585408,
    15.3297097168, 15.3622914957, 15.3948043183, 15.4272486205,
    15.4596248337, 15.4919333848, 15.5241746963, 15.5563491861,
    15.5884572681, 15.6204993518, 15.6524758425, 15.6843871414,
    15.7162336455, 15.748015748, 15.7797338381, 15.8113883008,
    15.8429795178, 15.8745078664, 15.9059737206, 15.9373774505,
    15.9687194227,
};
```

May 16, 04 20:23 **eg58−const.c** Page 1/1

```
/*
 * Introducing another modifier "const", which
 * tells the compiler that the variable should
 * stay constant and should not be modified.
 */
int main()
{
    const int i = 0;
    i = 1;
    return 0;
}
```

May 16, 04 20:44 **eg59−Makefile** Page

```
# This makefile includes
# − command to force gcc to link statically with a library
# − commands to produce a dynamic library (−shared) and
#   commands to produce object files that are suitable
#   for dynamic library (−fPIC)
# − commands to compile with profiling information (−pg)
#   and how to invoke gprof to profile your code.

edge_dynamic: edge.c
    gcc −o edge_dynamic edge.c −lm

edge_static: edge.c
    gcc −static −o edge_static edge.c −lm

CC=gcc
CFLAGS=−fPIC

lib2281.so: eg39−module.o
    $(CC) −shared −o lib2281.so eg39−module.o

main: eg39−main.o
    $(CC) eg39−main.c −L. −o main −l2281

profile:
    gcc −pg −g −o edge edge.c −lm
    edge in.pgm > out.pgm
    gprof edge | less
```