

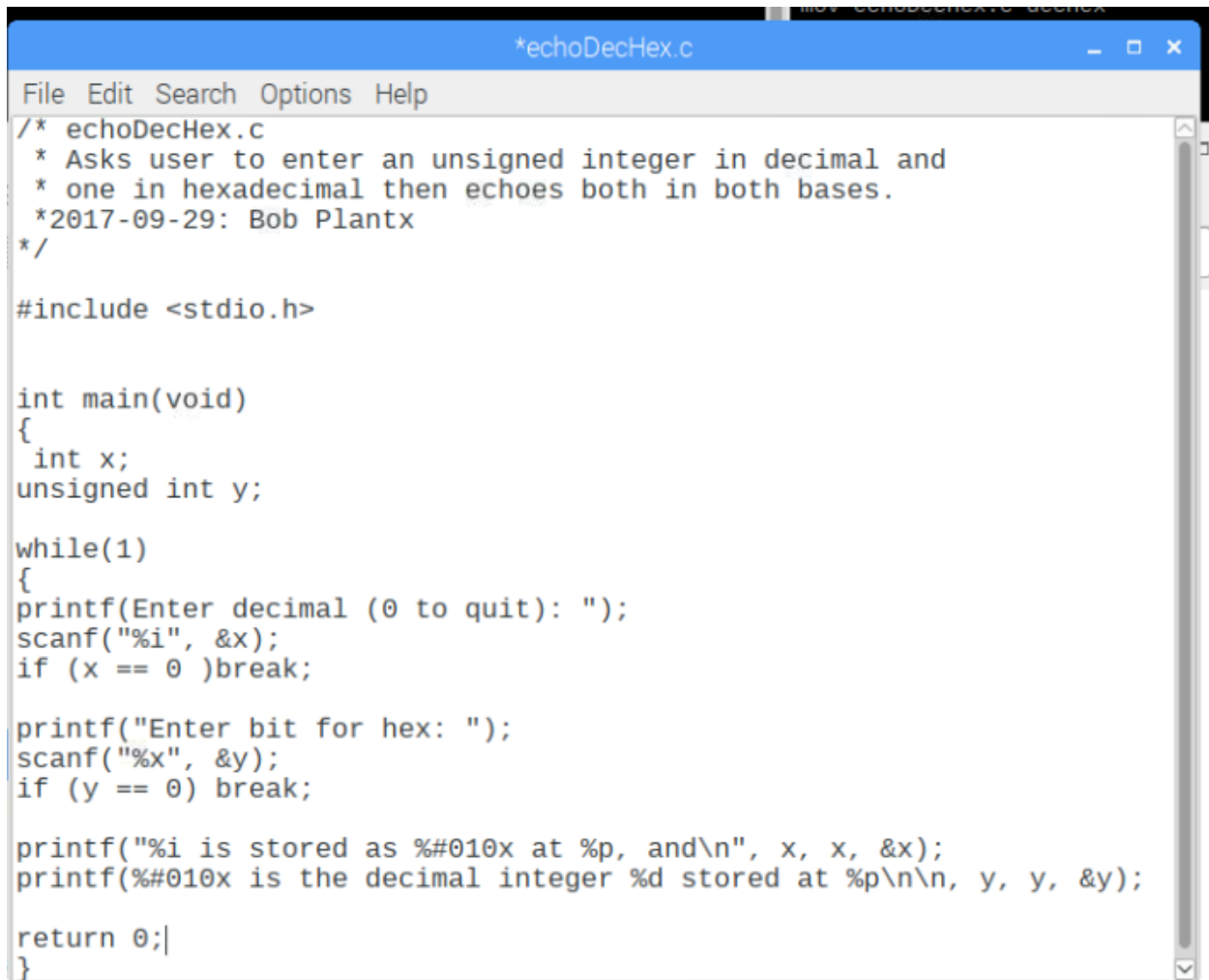
```
dec
File Edit Tabs Help
Enter an unsigned decimal integer: 100
Enter a bit pattern in hexadecimal: 64
100 is stored as 0x00000064, and
0x00000064 represents the unsigned decimal integer 100
Enter an unsigned decimal integer: 88
Enter a bit pattern in hexadecimal: 58
88 is stored as 0x00000058, and
0x00000058 represents the unsigned decimal integer 88
Enter an unsigned decimal integer: 32768
Enter a bit pattern in hexadecimal: 8000
32768 is stored as 0x00008000, and
0x00008000 represents the unsigned decimal integer 32768
Enter an unsigned decimal integer: 43981
Enter a bit pattern in hexadecimal: abcd
43981 is stored as 0x0000abcd, and
0x0000abcd represents the unsigned decimal integer 43981
Enter an unsigned decimal integer: 
```

2. largest unsigned integer is 32 bits

```
0x0000abcd represents the unsigned decimal integer 43981
Enter an unsigned decimal integer: 2342
Enter a bit pattern in hexadecimal: ffffffffffffffff
2342 is stored as 0x00000926, and
0xffffffff represents the unsigned decimal integer 4294967295
Enter an unsigned decimal integer: 232
Enter a bit pattern in hexadecimal: ffffffff
232 is stored as 0x000000e8, and
0xffffffff represents the unsigned decimal integer 4294967295
Enter an unsigned decimal integer: 
```

3. 7fffffff as greatest integer before breaking

```
0x07fffffff represents the unsigned decimal integer 134217727
Enter an unsigned decimal integer: 7fffffff
Enter a bit pattern in hexadecimal: 7 is stored as 0x00000007, and
0x07fffffff represents the unsigned decimal integer 268435455
Enter an unsigned decimal integer: 21
```



```
File Edit Search Options Help
/* echoDecHex.c
 * Asks user to enter an unsigned integer in decimal and
 * one in hexadecimal then echoes both in both bases.
 *2017-09-29: Bob Plantx
 */

#include <stdio.h>

int main(void)
{
    int x;
    unsigned int y;

    while(1)
    {
        printf("Enter decimal (0 to quit): ");
        scanf("%i", &x);
        if (x == 0 )break;

        printf("Enter bit for hex: ");
        scanf("%x", &y);
        if (y == 0) break;

        printf("%i is stored as %#010x at %p, and\n", x, x, &x);
        printf("%#010x is the decimal integer %d stored at %p\n\n", y, y, &y);

        return 0;
    }
}
```

1.

```
pi@raspberrypi:~/Desktop/code $ cd 2.9.1
pi@raspberrypi:~/Desktop/code/2.9.1 $ cc -g intAndFloat.c
pi@raspberrypi:~/Desktop/code/2.9.1 $ gdb a.out
GNU gdb (Raspbian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "arm-linux-gnueabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...done.
(gdb) run
Starting program: /home/pi/Desktop/code/2.9.1/a.out
The integer is 19088743 and the float is 19088.742188
Inferior 1 (process 28639) exited normally
(gdb) █
```

2.14

```
GNU nano 2.7.4 File: stringInHex.c
#include <stdio.h>

int main(void)
{
    char *stringPtr = "Hello world. \n";
    while (*stringPtr != '\0')
    {
        printf("%p: ", stringPtr);
        printf("0x%02x\n", *stringPtr);
        stringPtr++;
    }
    printf("%p: ", stringPtr);
    printf("0x%02x\n", *stringPtr);

    return 0;
}

[ File 'stringInHex.c' is unwritable ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```