Bit Manipulation:

Program #1: Turning on a range on bits starting at a particular bit position:

Startbit represents the starting bit position and numbits represents the number of bits to be turned on. In this case, bits that are already set stay set.

oid main()

{

unsigned short a=31, c;

int numbits, startbit;

numbits=3; startbit=1;

c= a | (~((unsigned short)~0 <<numbits)<<startbit);

printf("c=%hu\n",c);

}

Program #2: Turning off a range off bits starting at a particular bit position:

Startbit represents the starting bit position and numbits represents the number of bits to be turned off. In this case, bits that are already unset stay unset.

void main()

{

unsigned short a=31, c;

int numbits, startbit;

numbits=3; startbit=1;

c= a & ~(~((unsigned short)~0 <<numbits)<<startbit);

printf("c=%hu\n",c);

}

Program #3: Toggle bits starting at a particular bit position:

void main()

{

unsigned short a=13, c;

int numbits, startbit;

numbits=3; startbit=1;

c= a ^ (~((unsigned short)~0 <<numbits)<<startbit);

printf("c=%hu\n",c);

}

Program #4: Rotation of bits

Concept: A left rotation of n bits results in the same value as a right rotation by 16-n bits.In either case both rotation needs to be done. This is to cover the bits in the end

#include<stdio.h>

#include<string.h>

void main()

{

//unsigned short a=3;

unsigned short word=3;

unsigned short rightrot(unsigned short word, int numbits);

word=rightrot(word,1);

printf("%hu\n",word);

}

unsigned short rightrot(unsigned short word, int numbits)

{

unsigned short numleft;

numleft = word << (sizeof(short)\*8-numbits); //left rotation by 16-n in this case, since right rotation is desired.

word = word >>numbits; //right rotation.

word=word | numleft;

return word;

}

Program #5: Check if the bit at a particular position is set or unset.

Concept: If you want to check if bit at position 2 is set, move the bits by 2 positions to the right. Now bit at position 2 is at position 0. Do AND operation with word and 1, if return value is 1 then bit is set else bit is not set.

Unsigned short setbit(unsigned short word, int i)

{

word=word >> i;

word=word&1;

Return word;

}