**BITWISE**

**COPY BITS**

#include <stdio.h>

#define COPY (((snum << (8 - s - 1)) >> (8 - n)) << (d - n + 1)) | ((dnum << (d + 1)) >> (d + 1)) | ((dnum << (d - n + 1)) >> (d - n + 1))

void bitwise\_display(unsigned int num);

int main()

{

unsigned int snum;

unsigned int dnum;

unsigned int n;

unsigned int s;

unsigned int d;

printf("Enter the first number: \n");

scanf("%d", &snum);

printf("Enter the second number: \n");

scanf("%d", &dnum);

printf("Enter the number of bits to be copied: \n");

scanf("%d", &n);

printf("Enter the position in first number: \n");

scanf("%d", &s);

printf("Enter the position in second number: \n");

scanf("%d", &d);

printf("\nOriginal first number: ");

bitwise\_display(snum);

printf("Original second number: ");

bitwise\_display(dnum);

printf("\nModified second number: ");

bitwise\_display(COPY);

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**COUNT LEADING AND TRAILING BITS**

#include <stdio.h>

void bitwise\_display(unsigned int num);

unsigned int count\_leading\_set\_bits(unsigned int num);

unsigned int count\_leading\_clear\_bits(unsigned int num);

unsigned int count\_trailing\_set\_bits(unsigned int num);

unsigned int count\_trailing\_clear\_bits(unsigned int num);

int main()

{

unsigned int num;

printf("Enter the number: \n");

scanf("%d", &num);

printf("\nOriginal number: ");

bitwise\_display(num);

printf("Leading set bits count is %d\n", count\_leading\_set\_bits(num));

printf("Leading clear bits count is %d\n", count\_leading\_clear\_bits(num));

printf("Trailing set bits count is %d\n", count\_trailing\_set\_bits(num));

printf("Trailing clear bits count is %d\n", count\_trailing\_clear\_bits(num));

}

unsigned int count\_leading\_set\_bits(unsigned int num)

{

unsigned int count = 0;

for (int i = 0; i < 8; i++) {

if (num & 128) {

count++;

} else {

return count;

}

num <<= 1;

}

return count;

}

unsigned int count\_leading\_clear\_bits(unsigned int num)

{

unsigned int count = 0;

for (int i = 0; i < 8; i++) {

if ((num & 128) == 0) {

count++;

} else {

return count;

}

num <<= 1;

}

return count;

}

unsigned int count\_trailing\_set\_bits(unsigned int num)

{

unsigned int count = 0;

for (int i = 0; i < 8; i++) {

if (num & 1) {

count++;

} else {

return count;

}

num >>= 1;

}

return count;

}

unsigned int count\_trailing\_clear\_bits(unsigned int num)

{

unsigned int count = 0;

for (int i = 0; i < 8; i++) {

if ((num & 1) == 0) {

count++;

} else {

return count;

}

num >>= 1;

}

return count;

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**COUNT SET AND CLEAR BITS**

#include <stdio.h>

void bitwise\_display(unsigned int num);

unsigned int count\_set\_bits(unsigned int num);

unsigned int count\_clear\_bits(unsigned int num);

int main()

{

unsigned int num;

printf("Enter the number: \n");

scanf("%d", &num);

printf("\nOriginal number: ");

bitwise\_display(num);

printf("Set bits count is %d\n", count\_set\_bits(num));

printf("Clear bits count is %d\n", count\_clear\_bits(num));

}

unsigned int count\_set\_bits(unsigned int num)

{

unsigned int count = 0;

for (int i = 0; i < 8; i++) {

if (num & 1) {

count++;

}

num >>= 1;

}

return count;

}

unsigned int count\_clear\_bits(unsigned int num)

{

unsigned int count = 0;

for (int i = 0; i < 8; i++) {

if ((num & 1) == 0) {

count++;

}

num >>= 1;

}

return count;

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**GET BITS**

#include <stdio.h>

#define GET ((snum << (32 - (p + n))) >> (32 - n))

void bitwise\_display(unsigned char num);

int main()

{

unsigned int snum;

int p;

int n;

printf("Enter the number: \n");

scanf("%i", &snum);

printf("Enter the number of bits: \n");

scanf("%d", &n);

printf("Enter the position: \n");

scanf("%d", &p);

bitwise\_display(snum);

bitwise\_display(GET);

}

void bitwise\_display(unsigned char num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**INVERT BITS**

#include <stdio.h>

#include <math.h>

#define INVERT ((n1 << (p - n + 1)) ^ snum)

void bitwise\_display(unsigned char num);

int main()

{

unsigned char snum;

int p;

int n;

int n1 = 1;

printf("Enter the number: \n");

scanf("%hhd", &snum);

printf("Enter the number of bits: \n");

scanf("%d", &n);

for(int i = 0; i < n; i++) {

n1 = 2 \* n1;

}

n1 = n1 - 1;

printf("Enter the position: \n");

scanf("%d", &p);

bitwise\_display(snum);

bitwise\_display(INVERT);

}

void bitwise\_display(unsigned char num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**ROTATE BITS**

#include <stdio.h>

#define LEFT ((num << n) | (num >> (8 - n)))

#define RIGHT ((num >> n) | (num << (8 - n)))

void bitwise\_display(unsigned int num);

int main()

{

unsigned int num;

unsigned int num1;

unsigned int n;

printf("Enter the number: \n");

scanf("%d", &num);

printf("Enter the number of positions to rotate: ");

scanf("%d", &n);

printf("\nOriginal number: ");

bitwise\_display(num);

printf("Left rotate: ");

bitwise\_display(LEFT);

printf("Right rotate: ");

bitwise\_display(RIGHT);

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**SET BITS**

#include <stdio.h>

#define SET (((snum << (32 - n)) >> (32 - p - 1)) | ((dnum >> (p + 1)) << (p + 1)) | ((dnum << (32 - n - 1)) >> (32 - n - 1)))

void bitwise\_display(unsigned int num);

int main()

{

unsigned int snum;

unsigned int dnum;

unsigned int n;

unsigned int p;

printf("Enter the first number: \n");

scanf("%d", &snum);

printf("Enter the second number: \n");

scanf("%d", &dnum);

printf("Enter the number of bits to be copied: \n");

scanf("%d", &n);

printf("Enter the position in second number: \n");

scanf("%d", &p);

printf("\nOriginal first number: ");

bitwise\_display(snum);

printf("Original second number: ");

bitwise\_display(dnum);

printf("\nModified second number: ");

bitwise\_display(SET);

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**SWAP BITS BETWEEN**

#include <stdio.h>

void bitwise\_display(unsigned int num);

int swap\_bits\_between(unsigned int snum, unsigned int dnum, unsigned int s, unsigned int d);

int main()

{

unsigned int snum;

unsigned int dnum;

unsigned int s;

unsigned int d;

printf("Enter the first number: \n");

scanf("%d", &snum);

printf("Enter the second number: \n");

scanf("%d", &dnum);

printf("Enter the position to be interchanged in first number: ");

scanf("%d", &s);

printf("Enter the position to be interchanged in second number: ");

scanf("%d", &d);

printf("\nOriginal numbers: ");

bitwise\_display(snum);

bitwise\_display(dnum);

if (swap\_bits\_between(snum, dnum, s, d) == 0) {

printf("\nSuccessful\n");

} else {

printf("Bits are same\n");

}

}

int swap\_bits\_between(unsigned int snum, unsigned int dnum, unsigned int s, unsigned int d)

{

if (((snum >> s) & 1) != ((dnum >> d) & 1)) {

snum = ((1 << s) ^ snum);

dnum = ((1 << d) ^ dnum);

printf("\nModified numbers: ");

bitwise\_display(snum);

bitwise\_display(dnum);

return 0;

} else {

return -1;

}

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**SWAP BITS WITHIN**

#include <stdio.h>

void bitwise\_display(unsigned int num);

unsigned int swap\_bits\_within(unsigned int num, unsigned int s, unsigned int d);

int main()

{

unsigned int num;

unsigned int s;

unsigned int d;

printf("Enter the number: \n");

scanf("%d", &num);

printf("Enter the 2 positions to be interchanged: ");

scanf("%d %d", &s, &d);

bitwise\_display(num);

num = swap\_bits\_within(num, s, d);

bitwise\_display(num);

}

unsigned int swap\_bits\_within(unsigned int num, unsigned int s, unsigned int d)

{

if (((num >> s) & 1) != ((num >> d) & 1)) {

num = (((1 << s) | (1 << d)) ^ num);

}

return num;

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**TEST SET BITS**

#include <stdio.h>

#define TESTSET ((1 << p) & num) ? (num) : ((1 << p) | num)

void bitwise\_display(unsigned int num);

int main()

{

unsigned int num;

unsigned int p;

printf("Enter the number: \n");

scanf("%d", &num);

printf("Enter the position to set: ");

scanf("%d", &p);

printf("\nOriginal number: ");

bitwise\_display(num);

printf("Modified number: ");

bitwise\_display(TESTSET);

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**TOGGLE BITS**

#include <stdio.h>

#define EVEN (num ^ (1 | (1 << 2) | (1 << 4) | (1 << 6)))

#define ODD (num ^ ((1 << 1) | (1 << 3) | (1 << 5) | (1 << 7)))

void bitwise\_display(unsigned int num);

int main()

{

unsigned int num;

printf("Enter the number: \n");

scanf("%d", &num);

printf(" Original number: ");

bitwise\_display(num);

printf("Toggle even bits: ");

bitwise\_display(EVEN);

printf(" Toggle odd bits: ");

bitwise\_display(ODD);

}

void bitwise\_display(unsigned int num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}

**MACROS**

#include <stdio.h>

#define MAX for (i = 7; i >= 0; i--) { \

if ((num1 & (1 << i)) && (!(num2 & (1 << i)))) { \

printf("Num1 is greater\n"); \

break;\

} else if (!(num1 & (1 << i)) && ((num2 & (1 << i)))) { \

printf("Num2 is greater\n"); \

break;\

} \

} \

if (i < 0) { \

printf("Numbers are equal\n"); \

}

void bitwise\_display(unsigned char num);

int main()

{

unsigned char num1;

unsigned char num2;

int p;

int n;

int i;

printf("Enter the number1: \n");

scanf("%hhd", &num1);

printf("Enter the number2: \n");

scanf("%hhd", &num2);

printf("Num1: \n");

bitwise\_display(num1);

printf("Num2: \n");

bitwise\_display(num2);

MAX

}

void bitwise\_display(unsigned char num)

{

int i;

for (i = 0; i < 8; i++) {

if (num & 128) {

printf("1 ");

} else {

printf("0 ");

}

num <<= 1;

}

printf("\n");

}