**Problem 1**

**Code –**

#include <stdio.h>

#include <stdlib.h>

#include <stdint.h> // For int8\_t

int main(){

unsigned short int hex[8];

hex[0] = 0x0A;

hex[1] = 0x06;

hex[2] = 0x14;

hex[3] = 0x6B;

hex[4] = 0x8A;

hex[5] = 0x86;

hex[6] = 0x94;

hex[7] = 0xEB;

printf("%10s %10s %10s %10s\n", "Hex", "Binary", "Unsigned", "Signed");

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

printf("%10X ", hex[i]);

printf(" ");

for(int j = 7; j >= 0; j--){

printf("%d", (hex[i] >> j) & 1);

}

printf("%10d ", hex[i]);

int sign = (int)((int8\_t)(hex[i] & 0xFF));

printf("%10d\n", sign);

}

}

A computer screen with white text

AI-generated content may be incorrect.

|  |  |  |  |
| --- | --- | --- | --- |
| **Hexa-Decimal** | **Binary** | **B2U8(x)** | **B2T8(x)** |
| 0x0A | 00001010 | 10 | 10 |
| 0x06 | 00000110 | 6 | 6 |
| 0x14 | 00010100 | 20 | 20 |
| 0x6B | 01101011 | 107 | 107 |
| 0x8A | 10001010 | 138 | -118 |
| 0x86 | 10000110 | 134 | -122 |
| 0x94 | 10010100 | 148 | -108 |
| 0xEB | 11101011 | 235 | -21 |

**Problem 2**

**Code –**

#include <stdio.h>

#include <limits.h>

#include <string.h>

void print\_line(int x){

for (int i = 0; i < x; i++) printf("-");

printf("\n");

}

int main(){

// Print header center align

int width = 100;

char text[] = "Word size w";

int padding = (width - strlen(text)) / 2;

printf("%\*s%s%\*s\n", padding, "", text, padding, "");

print\_line(80);

// Print second row 'Value 8 16 32 64'

printf("%-10s %-8d %-8d %-20d %-40d\n", "Value", 8, 16, 32, 64);

print\_line(80);

int w[4] = {8, 16, 32, 64};

// Print Unsigned max

unsigned long long max = ULLONG\_MAX;

printf("%-10s", "U Max(w)");

printf("0x%-8llX", (max >> (64 - 8)));

printf("0x%-8llX", (max >> (64 - 16)));

printf("0x%-20llX", (max >> (64 - 32)));

printf("0x%-40llX\n", (max >> (64 - 64)));

printf("%-10s", "");

printf("%-10llu", (max >> (64 - 8)));

printf("%-10llu", (max >> (64 - 16)));

printf("%-22llu", (max >> (64 - 32)));

printf("%-42llu\n\n", (max >> (64 - 64)));

// Print Signed min

long long min = LLONG\_MIN;

printf("%-10s", "T Min(w)");

printf("0x%-8X", (min >> (64 - 8)) & 0xFF);

printf("0x%-8X", (min >> (64 - 16)) & 0xFFFF);

printf("0x%-20X", (min >> (64 - 32)) & 0xFFFFFFFF);

printf("0x%-40llX\n", (min >> (64 - 64)) & 0xFFFFFFFFFFFFFFFF);

printf("%-10s", "");

printf("%-10d", (min >> (64 - 8)));

printf("%-10d", (min >> (64 - 16)));

printf("%-22lld", (min >> (64 - 32)));

printf("%-42lld\n\n", (min >> (64 - 64)));

// Print Signed max

long long signed\_max = LLONG\_MAX;

printf("%-10s", "T Max(w)");

printf("0x%-8llX", (signed\_max >> (64 - 8)));

printf("0x%-8llX", (signed\_max >> (64 - 16)));

printf("0x%-20llX", (signed\_max >> (64 - 32)));

printf("0x%-40llX\n", (signed\_max >> (64 - 64)));

printf("%-10s", "");

printf("%-10lld", (signed\_max >> (64 - 8)));

printf("%-10lld", (signed\_max >> (64 - 16)));

printf("%-22lld", (signed\_max >> (64 - 32)));

printf("%-42lld\n\n", (signed\_max >> (64 - 64)));

// Print -1

long long neg\_one = -1;

printf("%-10s", "-1");

printf("0x%-8llX", (neg\_one >> (64 - 8)) & 0xFF);

printf("0x%-8llX", (neg\_one >> (64 - 16)) & 0xFFFF);

printf("0x%-20llX", (neg\_one >> (64 - 32)) & 0xFFFFFFFF);

printf("0x%-40llX\n\n", (neg\_one >> (64 - 64)) & 0xFFFFFFFFFFFFFFFF );

// Print 0

long long zero = 0;

printf("%-10s", "0");

char col2[10];

snprintf(col2, sizeof(col2), "0x%02llX", (zero >> (64 - 8)) & 0xFF);

char col3[12];

snprintf(col3, sizeof(col3), "0x%04llX", (zero >> (64 - 16)) & 0xFFFF);

char col4[16];

snprintf(col4, sizeof(col4), "0x%08llX", (zero >> (64 - 32)) & 0xFFFFFFFF);

char col5[22];

snprintf(col5, sizeof(col5), "0x%016llX", (zero >> (64 - 64)) & 0xFFFFFFFFFFFFFFFF);

printf("%-10s", col2);

printf("%-10s", col3);

printf("%-22s", col4);

printf("%-42s\n", col5);

print\_line(80);

return 0;

}

A screenshot of a computer program

AI-generated content may be incorrect.

**Problem 3**

**Code –**

#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

int main(){

printf("%10s %10s %10s\n", "Binary", "Unsigned", "Signed");

for(unsigned short int i = 0; i < 32; i++){

printf(" ");

for(int j = 4; j >= 0; j--){

printf("%d", (i >> j) & 1);

}

printf("%10d", i);

int sign = ((i & 0x1F) << 27) >> 27;

printf("%10d\n", sign);

}

}

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Problem 4**

**Code –**

#include <limits.h>

#include <stdio.h>

int main(){

int signed\_1 = 0, signed\_2;

unsigned int unsigned\_1 = 0, unsigned\_2;

printf("%15s%20s%10s%15s\n", "Constant 1", "Constant 2", "Relation", "True or False");

// 0 == 0U

if(signed\_1 == unsigned\_1){

printf("%15d%19uU%10s%15s\n", signed\_1, unsigned\_1, "==", "True");

} else {

printf("%15d%19uU%10s%15s\n", signed\_1, unsigned\_1, "==", "False");

}

signed\_2 = -1;

// -1 < 0

if(signed\_2 < signed\_1){

printf("%15d%20d%10s%15s\n", signed\_2, signed\_1, "<", "True");

} else {

printf("%15d%20d%10s%15s\n", signed\_2, signed\_1, "<", "False");

}

// -1 > 0U

if(signed\_2 > unsigned\_1){

printf("%15d%19dU%10s%15s\n", signed\_2, unsigned\_1, ">", "True");

} else {

printf("%15d%19dU%10s%15s\n", signed\_2, unsigned\_1, ">", "False");

}

signed\_1 = INT\_MAX;

signed\_2 = INT\_MIN;

// Tmax > TMin

if(signed\_1 > signed\_2){

printf("%15d%20d%10s%15s\n", signed\_1, signed\_2, ">", "True");

} else {

printf("%15d%20d%10s%15s\n", signed\_1, signed\_2, ">", "False");

}

unsigned\_1 = (unsigned int) INT\_MAX;

// TMaxU > TMin

if(unsigned\_1 < signed\_2){

printf("%14uU%20d%10s%15s\n", unsigned\_1, signed\_2, "<", "True");

} else {

printf("%14uU%20d%10s%15s\n", unsigned\_1, signed\_2, "<", "False");

}

signed\_1 = -1, signed\_2 = -2;

// -1 > -2

if(signed\_1 > signed\_2){

printf("%15d%20d%10s%15s\n", signed\_1, signed\_2, ">", "True");

} else {

printf("%15d%20d%10s%15s\n", signed\_1, signed\_2, ">", "False");

}

// (unsigned)-1 > -2

if((unsigned int)signed\_1 > signed\_2){

printf(" (unsigned)%-7d%15d%10s%15s\n", (unsigned int)signed\_1, signed\_2, ">", "True");

} else {

printf(" (unsigned)%-7d%15d%10s%15s\n", (unsigned int)signed\_1, signed\_2, ">", "False");

}

signed\_1 = INT\_MAX;

unsigned\_1 = (unsigned int)INT\_MAX + 1;

// TMax > TMax+1(unsigned int)

if (signed\_1 < unsigned\_1) {

printf("%15d%19uU%10s%15s\n", signed\_1, unsigned\_1, "<", "True");

} else {

printf("%15d%19uU%10s%15s\n", signed\_1, unsigned\_1, "<", "False");

}

// TMax > (int) TMax(unsigned int)

if (signed\_1 > (int)unsigned\_1) {

printf("%15d (int)%-10dU%10s%15s\n", signed\_1, (int)unsigned\_1, ">", "True");

} else {

printf("%15d (int)%15dU%10s%15s\n", signed\_1, (int)unsigned\_1, ">", "False");

}

}

A computer screen with white text

AI-generated content may be incorrect.

**Problem 5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type** | **x** | **y** | **x+y** | **x+(t5)y** | **Case** |
| integer | 13 | 5 | 18 | 18 | 4 |
| binary | 01101 | 00101 | 10010 | 10010 |
| integer | 3 | 4 | 7 | 7 | 3 |
| binary | 00011 | 00100 | 00111 | 00111 |
| integer | 24 | 7 | 31 | 31 | 4 |
| binary | 11000 | 00111 | 11111 | 11111 |
| integer | 23 | 25 | 48 | 16 | 4 |
| binary | 10111 | 11001 | 110000 | 10000 |
| integer | 21 | 18 | 39 | 7 | 3 |
| binary | 10101 | 10010 | 100111 | 00111 |

**Problem 6**

**Code –**

#include <stdio.h>

#include <limits.h>

#include <math.h>

void print\_bits(int x){

for(int i = (sizeof(int) \* 8 - 1); i >= 0; i--) printf("%d", (x >> i) & 1);

printf("\n");

}

int saturating\_add(int x, int y){

unsigned int sum = x + y, w = sizeof(int) \* 8;

unsigned int msb\_x = (x >> (w - 1)) & 1, msb\_y = (y >> (w - 1)) & 1, msb\_sum = (sum >> (w - 1)) & 1;

int positive\_overflow = ~msb\_x & ~msb\_y & msb\_sum;

int negative\_overflow = msb\_x & msb\_y & ~msb\_sum;

int result = (sum & ~(-positive\_overflow | -negative\_overflow)) | (-positive\_overflow & INT\_MAX) | (-negative\_overflow & INT\_MIN);

return result;

}

int main(){

// Positive overflow

printf("INT\_MAX + 5 = %d\n", saturating\_add(INT\_MAX, 5));

// Negative overflow

printf("INT\_MIN + -5 = %d\n", saturating\_add(INT\_MIN, -5));

// Normal

printf("5 + 5 = %d\n", saturating\_add(5, 5));

}

**A black background with white text

AI-generated content may be incorrect.**