

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);
    }
}
```

```
● adirathodd@adi HW2 % gcc-14 p1.c -o p1
● adirathodd@adi HW2 % ./p1
      Hex      Binary  Unsigned  Signed
        A    00001010         10        10
         6    00000110          6          6
        14    00010100         20         20
        6B    01101011        107        107
        8A    10001010        138       -118
        86    10000110        134       -122
        94    10010100        148       -108
        EB    11101011        235        -21
○ adirathodd@adi HW2 %
```

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%-8lX", (max >> (64 - 8)));
printf("0x%-8lX", (max >> (64 - 16)));
printf("0x%-20lX", (max >> (64 - 32)));
printf("0x%-40lX\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%-40lX\n", (min >> (64 - 64)) & 0xFFFFFFFFFFFFFFFF);

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

printf("%-10d", (min >> (64 - 8)));
printf("%-10d", (min >> (64 - 16)));
printf("%-22ld", (min >> (64 - 32)));
printf("%-42ld\n\n", (min >> (64 - 64)));

// Print Signed max
long long signed_max = LLONG_MAX;

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

printf("0x%-8lX", (signed_max >> (64 - 8)));
printf("0x%-8lX", (signed_max >> (64 - 16)));
printf("0x%-20lX", (signed_max >> (64 - 32)));
printf("0x%-40lX\n", (signed_max >> (64 - 64)));

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

printf("%-10lld", (signed_max >> (64 - 8)));
printf("%-10lld", (signed_max >> (64 - 16)));

```

```

printf("%-22ld", (signed_max >> (64 - 32)));
printf("%-42ld\n\n", (signed_max >> (64 - 64)));

// Print -1
long long neg_one = -1;

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

printf("0x%-8lX", (neg_one >> (64 - 8)) & 0xFF);
printf("0x%-8lX", (neg_one >> (64 - 16)) & 0xFFFF);
printf("0x%-20lX", (neg_one >> (64 - 32)) & 0xFFFFFFFF);
printf("0x%-40lX\n\n", (neg_one >> (64 - 64)) & 0xFFFFFFFFFFFFFFFF);

// Print 0
long long zero = 0;

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

char col2[10];
snprintf(col2, sizeof(col2), "0x%02lX", (zero >> (64 - 8)) & 0xFF);

char col3[12];
snprintf(col3, sizeof(col3), "0x%04lX", (zero >> (64 - 16)) & 0xFFFF);

char col4[16];
snprintf(col4, sizeof(col4), "0x%08lX", (zero >> (64 - 32)) & 0xFFFFFFFF);

char col5[22];
snprintf(col5, sizeof(col5), "0x%016lX", (zero >> (64 - 64)) & 0xFFFFFFFFFFFFFFFF);

printf("%-10s", col2);
printf("%-10s", col3);
printf("%-22s", col4);
printf("%-42s\n", col5);

print_line(80);

return 0;
}

```

adirathodd@adi HW2 % ./p2

	Word size w			
Value	8	16	32	64
U Max(w)	0xFF 255	0xFFFF 65535	0xFFFFFFFF 4294967295	0xFFFFFFFFFFFFFFFF 18446744073709551615
T Min(w)	0x80 -128	0x8000 -32768	0x80000000 -2147483648	0x8000000000000000 -9223372036854775808
T Max(w)	0x7F 127	0x7FFF 32767	0x7FFFFFFF 2147483647	0x7FFFFFFFFFFFFFFF 9223372036854775807
-1	0xFF	0xFFFF	0xFFFFFFFF	0xFFFFFFFFFFFFFFFF
0	0x00	0x0000	0x00000000	0x0000000000000000

adirathodd@adi HW2 %

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);
    }
}
```

● adirathodd@adi HW2 % ./p3

Binary	Unsigned	Signed
00000	0	0
00001	1	1
00010	2	2
00011	3	3
00100	4	4
00101	5	5
00110	6	6
00111	7	7
01000	8	8
01001	9	9
01010	10	10
01011	11	11
01100	12	12
01101	13	13
01110	14	14
01111	15	15
10000	16	-16
10001	17	-15
10010	18	-14
10011	19	-13
10100	20	-12
10101	21	-11
10110	22	-10
10111	23	-9
11000	24	-8
11001	25	-7
11010	26	-6
11011	27	-5
11100	28	-4
11101	29	-3
11110	30	-2
11111	31	-1

○ adirathodd@adi HW2 %

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");
}
}

```



```

● adirathodd@adi HW2 % ./p4
    Constant 1      Constant 2  Relation  True or False
      0          0U      ==      True
     -1          0      <      True
     -1         0U      >      True
  2147483647    -2147483648    >      True
 2147483647U   -2147483648    <      True
      -1         -2      >      True
(unsigned)-1   -2      >      True
  2147483647   2147483648U    <      True
 2147483647 (int)-2147483648U    >      True
○ adirathodd@adi HW2 % █

```

Problem 5

Type	x	y	x+y	x+(t5)y	Case
integer	13	5	18	-14	4
binary	01101	00101	010010	10010	
integer	3	4	7	7	3
binary	00011	00100	000111	00111	
integer	-8	7	31	-1	2
binary	11000	00111	011111	11111	
integer	-9	-7	-16	-16	2
binary	10111	11001	110000	10000	
integer	-11	-14	-25	7	1
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;

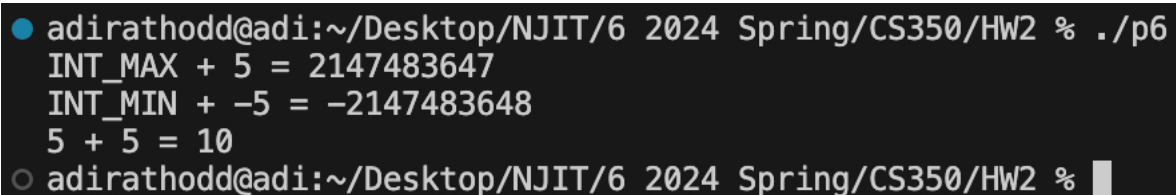
    int result = (sum & ~(-positive_overflow | -negative_overflow)) | (-positive_overflow & INT_MAX) | (-negative_overflow & INT_MIN);

    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));
}
```



```
● adirathodd@adi:~/Desktop/NJIT/6 2024 Spring/CS350/HW2 % ./p6
INT_MAX + 5 = 2147483647
INT_MIN + -5 = -2147483648
5 + 5 = 10
○ adirathodd@adi:~/Desktop/NJIT/6 2024 Spring/CS350/HW2 %
```