



Homework 2

For all questions assume following sizes

Int – 32 bit

Short int – 16 bits

Char – 8 bits



Question on Truncation (higher bits to lower bits)

Question 1: what will be printed values?

```
int i = -3;  
unsigned short u;  
u = i;  
printf( "%u", u );  
printf( "%d", u );
```





Solution

`int i = -3;` i will be 1...1 1101 (total 32 bits)

`unsigned short u;`

`u = i;` u will be 1...1 1101 (total 16 bits)

`printf("%u", u);` u will get integer promoted in 32 bits (source unsigned)

000...0 11...1101

`printf("%d", u);` u will get integer promoted in 32 bits (source unsigned)

000...0 11...1101

000...0 11...1101

this number is 65533 in both signed and unsigned hence 65533 will get printed in both



Question on Extension (lower bits to higher bits)

(Integer promotion based extension)

Question2: what will be printed values in the code below?

```
signed short ix = -3;  
printf( "%u", ix );  
printf( "%d", ix );
```



Solution

```
signed short ix = -3;
```

ix will be 1...1 1101 (total 16 bits)

```
printf( "%u", ix );
```

ix will be integer promoted (source signed)

111...1 11...1101

```
printf( "%d", ix );
```

ix will be integer promoted (source signed)

111...1 11...1101

111...1 11...1101

signed

this number is huge number ($2^{32} - 1 - 2$) in unsigned and -3 in

Answer: First printf will print huge number and 2nd print will print -3





Question3:

Consider k bit binary pattern.

T_{\max} and T_{\min} are maximum and minimum signed numbers we can represent using k bits.

U_{\max} and U_{\min} are maximum and minimum unsigned numbers we can represent using k bits.

Question 3.1

Question 3.2

Question 3.3

See Next Page





Question 3.1.

Given a binary pattern, which has value T_{\max} when interpreted as signed number. Find out the value of same pattern if interpreted as unsigned number

- A. T_{\max} B. U_{\max} C. $U_{\max}-1$ D. $T_{\max}+1$

Question 3.2.

Given a binary pattern, which has value T_{\min} when interpreted as signed number. Find out the value of same pattern if interpreted as unsigned number

- A. T_{\max} B. U_{\max} C. $U_{\max}-1$ D. $T_{\max}+1$

Question 3.3.

Given a binary pattern, which has value -1 when interpreted as signed number. Find out the value of same pattern if interpreted as unsigned number

- A. T_{\max} B. U_{\max} C. $U_{\max}-1$ D. $T_{\max}+1$



Solution:

3.1 – A

3.2 – D

3.3 – B

Positive values remain unchanged from 2's comp to unsigned. For example – 0101 is 5 in 2's complement then it is 5 in unsigned too.

Figure on right side shows the ranges of 2's compl and unsigned. And it also shows how mapping of positive integers in 2's compl happening for unsigned.

Tmax is mapping to Tmax and 0 is mapping to 0. It means there is no change for positive numbers in 2's compl range

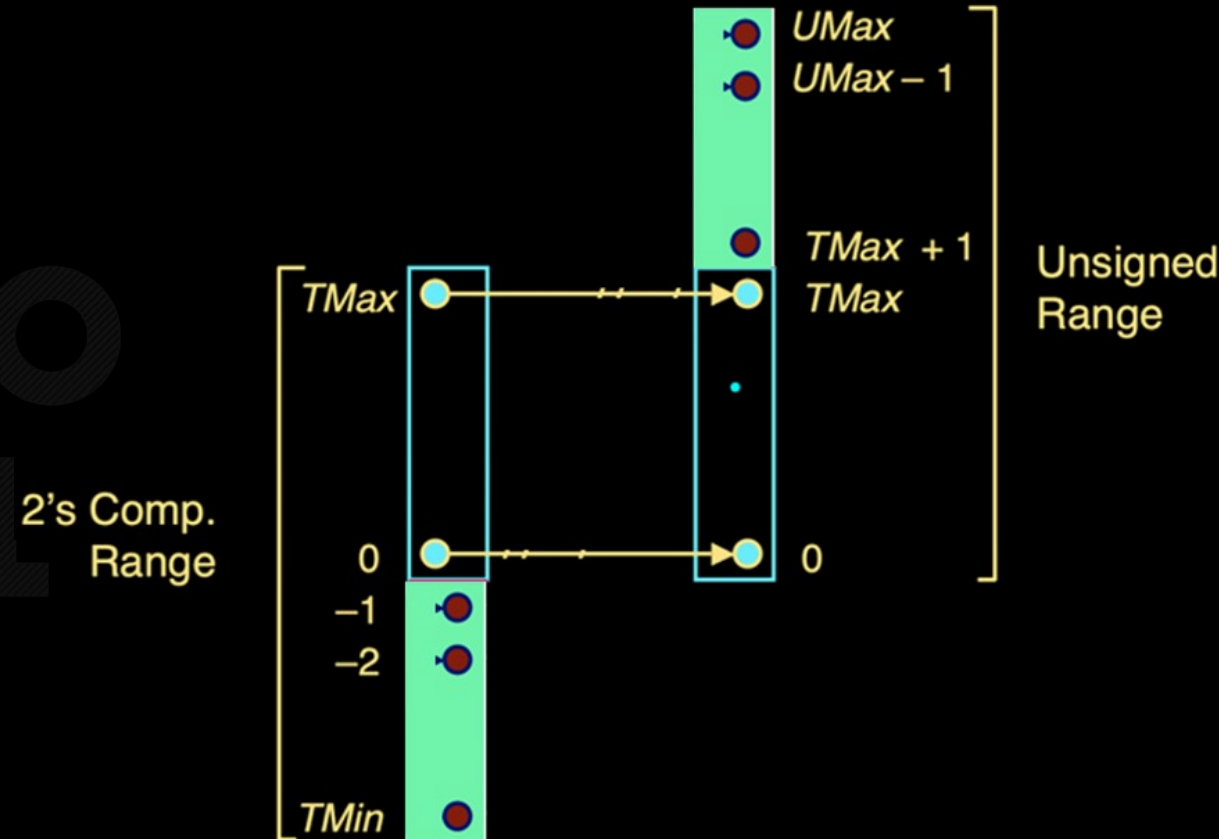
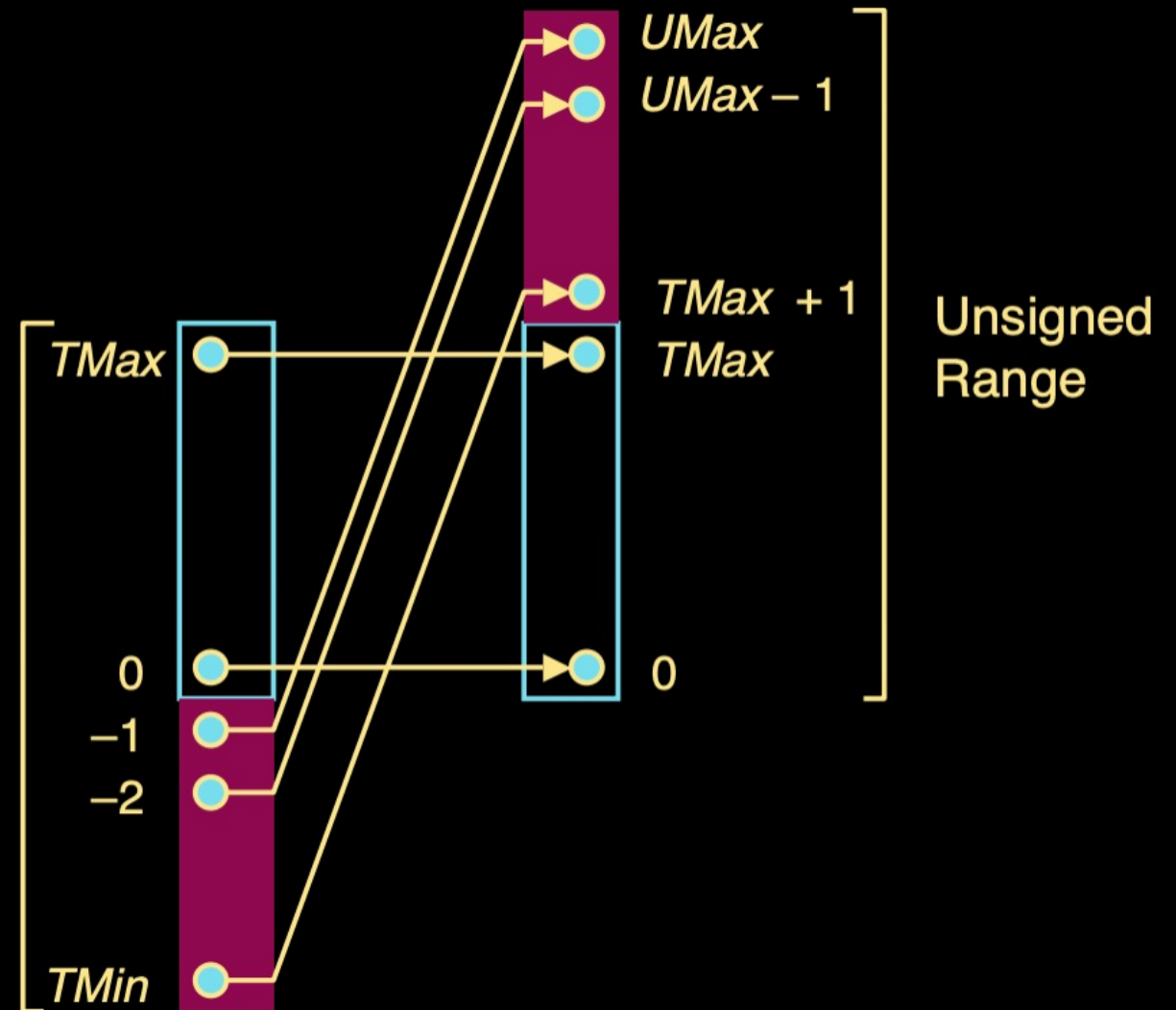


Figure on right side shows the complete mapping of all numbers.

T_{min} is $100\dots 0$, when we treat this number as unsigned then it will be $T_{max}+1$

2's Comp.
Range





Question 4

What will be the output of below program segment ?

```
a = 100;  
b = 200;  
c = a+b;  
printf("%d %d %d\n", a,b,c);
```

- A. 100 200 300
- B. 100 200 44
- C. 100 -56 44
- D. Can't say





Answer D

It depends on how we initialize a and b.
depending on we will have answer.



Question 5:

What will be the output of below program segment ?

```
signed char a = 100;  
signed char b = 200;  
signed int c = a+b;  
printf("%d %d %d\n", a,b,c);
```

- A. 100 200 300
- B. 100 200 44
- C. 100 -56 44
- D. Can't say



Solution

```
signed char a = 100;
```

a will be 8 bits `0110 0100`

```
signed char b = 200;
```

b will be 8 bits `1100 1000`

```
signed int c = a+b;
```

first a and b both will be converted to int because of integer promotion. In both source is signed.

a in 32 will be `00...0 0110 0100`
which is 100 in decimal

b in 32 will be `11...1 1100 1000` which is
same as 100 1000 (ignoring leading one's)
Which is $-2^6 + 2^3 = -64 + 8 = -56$

$C = 100 - 56 = 44$

```
printf("%d %d %d\n", a,b,c); // 100 -56 44 will be printed
```

(a and b will be integer promoted first)





Question 6

What will be the output of below program segment ?

```
signed char a = 100;  
signed char b = 200;  
signed char c = a+b;  
printf("%d %d %d\n", a,b,c);
```

- A. 100 200 300
- B. 100 200 44
- C. 100 -56 44
- D. Can't say





Solution: Same as question 5

```
signed char a = 100;
```

a will be 8 bits `0110 0100`

```
signed char b = 200;
```

b will be 8 bits `1100 1000`

```
signed char c = a+b;
```

first a and b both will be converted to int because of integer promotion. In both source is signed.

a in 32 will be `00...0 0110 0100`
which is 100 in decimal

b in 32 will be `11...1 1100 1000` which is
same as 100 1000 (ignoring leading one's)
Which is $-2^6 + 2^3 = -64 + 8 = -56$

$C = 100 - 56 = 44$

```
printf("%d %d %d\n", a,b,c); // 100 -56 44 will be printed  
(a, b and c will be integer promoted before printing)
```





Question 7

```
signed char a = 100;  
signed char b = 3;  
signed char c = 4;  
signed char result = a*b / c;
```

```
printf("%d", result);
```





Solution

```
signed char a = 100;  
signed char b = 3;  
signed char c = 4;  
signed char result = a*b / c;
```

* and / are left to right operators with same precedence
(will study in details in next class)
a*b will happen first as integer multiplication. Which is 300.
 $300/4 = 75$

```
printf("%d", result);    // 75 is output
```





Question 8

Fill one of the value below in method column of given table

- Sign Extension
- Zero Extension
- Truncation
- No truncation or extension required

From	To	Method
unsigned char	signed char	
unsigned char	signed short int	
signed char	signed int	
unsigned int	signed char	
signed short int	unsigned char	
Signed int	Signed short int	





Solution

1. Sign Extension
2. Zero Extension
3. Truncation
4. No truncation or extension required

From	To	Method
unsigned char	signed char	4
unsigned char	signed short int	2
signed char	signed int	1
unsigned int	signed char	3
signed short int	unsigned char	3
Signed int	Signed short int	3

