Today's plan

- Announcement:
  - Find your mid-term project partner and register your team on Piazza.
- Review
- Ex 5-3

Ex 5-2

Which of the following converts a string message (`char msg[]`, e.g. `"111101"`) to an integer (`int value`, e.g. 61)?

A
```
for (int i = value = 0; msg[i]; ++i)
    value = (value << 1) | (msg[i] - '0');
```

B
```
for (int i = value = 0; msg[i]; ++i)
    value |= (msg[i] - '0');
```

C
```
for (int i = value = 0; msg[i]; ++i)
    value = (2 * value + (msg[i] - '0'));
```

D
```
for (int i = value = 0; msg[i]; ++i)
    value *= 2 + (msg[i] - '0');
```

E
```
for (int i = value = 0; msg[i]; ++i)
    value = (msg[i] == '1') + (value << 1);
```

Number representations

Assume `sizeof(char) = 1`, `sizeof(short) = 2`, `sizeof(int) = 4` and `sizeof(long) = 8`. Also assume using two's complement representation. What are the ranges of `char`, `short`, `int`, and `long`?

A $[0, 2^8 - 1]$, $[0, 2^{16} - 1]$, $[0, 2^{32} - 1]$, $[0, 2^{64} - 1]$

B $[-2^4, 2^4]$, $[-2^8, 2^8]$, $[-2^{16}, 2^{16}]$, $[-2^{32}, 2^{32}]$

C $[-2^4, 2^4 - 1]$, $[-2^8, 2^8 - 1]$, $[-2^{16}, 2^{16} - 1]$, $[-2^{32}, 2^{32} - 1]$

D $[-2^4 + 1, 2^4]$, $[-2^8 + 1, 2^8]$, $[-2^{16} + 1, 2^{16}]$, $[-2^{32} + 1, 2^{32}]$

E $[-2^4 + 1, 2^4 - 1]$, $[-2^8 + 1, 2^8 - 1]$, $[-2^{16} + 1, 2^{16} - 1]$, $[-2^{32} + 1, 2^{32} - 1]$

Promotion, narrowing, and casting

What output is printed by the following program?

```c
#include <stdio.h>
int main(void) {
    int x = 20000;
    // 20,000 decimal is
    // 01001110 00100000
    // in binary
    printf("%d\n", (unsigned char) x);
    return 0;
}
```

A 0

B 16

C 20

D 32

E 20000

Promotion, narrowing, and casting

What output is printed by the following program?

```c
#include <stdio.h>
int main(void) {
    int n = 32065;
    float x = 24.79;
    printf("int n = %d but (char) n = %c\n", n, (char) n);
    printf("float x = %f but (long) x = %ld\n", x, (long) x);
    return 0;
}
```

A
```
int n = 32065 but (char) n = A
float x = 24.790001 but (long) x = 24
```

B
```
int n = 32065 but (char) n = A
float x = 24.79 but (long) x = 24
```

C
```
int n = 32065 but (char) n = A
float x = 24.790001 but (long) x = 25
```

D
```
int n = 32065 but (char) n = A
float x = 24.79 but (long) x = 25
```

E Unpredictable.

Promotion, narrowing, and casting

What output is printed by the following program?

```c
#include <stdio.h>

int main() {
    float b = 0.1;
    if (b == 0.1) printf("True\n");
    else printf("False\n");

    return 0;
}
```

A True

B False

Promotion, narrowing, and casting

What output is printed by the following program?

```c
#include <stdio.h>

int main() {
    double a = 0.1;
    float b = 0.1;
    double c = a / b;

    if (c == 1) printf("True\n");
    else printf("False\n");

    return 0;
}
```

A True

B False

Promotion, narrowing, and casting

What output is printed by the following program?

```c
#include <stdio.h>

int main() {
    unsigned int positive_1 = 1;
    int negative_1 = -1;

    if (positive_1 > negative_1) printf("True\n");
    else printf("False\n");

    return 0;
}
```

A True

B False

Promotion, narrowing, and casting

What output is printed by the following program?

```c
#include <stdio.h>

int main() {
    printf("%u, %u\n", UINT_MAX, UINT_MAX + 1);
    printf("INT_MAX + 1 > INT_MAX: %d\n", x + 1 > x);

    return 0;
}
```

A
```
4294967295, 0
INT_MAX + 1 > INT_MAX: 1
```

B
```
4294967295, 4294967296
INT_MAX + 1 > INT_MAX: 1
```

C
```
4294967295, 0
INT_MAX + 1 > INT_MAX: 0
```

D
```
4294967295, 4294967296
INT_MAX + 1 > INT_MAX: 0
```

E Unpredictable.

Promotion, narrowing, and casting

- Checkout the usual arithmetic conversion of C: `https://en.cppreference.com/w/c/language/conversion`. Notes:
  - wider data type is preferred.
  - floating point representation is preferred.
  - mixing signed and unsigned types require your attention.
- unsigned integer types are always wrapped. There is no overflow.
- signed integer types could have overflow, and it's an undefined behavior.
- However, 'gcc' optimizes some results. e.g. `INT_MAX + 1 > INT_MAX` is evaluated as 1.

Class exercises

Ex 5-3