# survey:

| Slang | Example | Meaning |
|---|---|---|
| Surely | “Surely it’s not that difficult” | It would be funny if this happened |
| Alky | “They brought their own alky” | Alcohol |
| Ayo | “Ayo?” | This piqued my interest |
| Chunder | “You better not chunder tonight” | Vomit |
| Slay | “Omg slay” | They are doing something really well |
| Shout | “It’s my shout” | Will pay the full price |
| Sus | “They sus” | Suspicious |
| Hits different | “That song hits different” | This is really good |
| Goes hard | “Why that pictures goes so hard tho” | This fits really well |
| Imagine | “Imagine losing” | This could never happen to me, unless ironically |
| As if | “As if you didn’t ask for help” | I can’t believe this didn’t occur |

# COMP1511 Week 3!

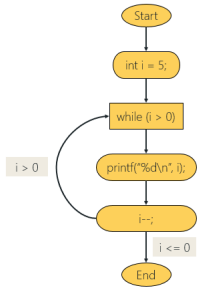
1D While Loops, 2D While Loops, Custom Types, Naming

# The Agenda

## While Loops (in 1D)

How does the code flow?

```
void a(void) {
    int i = 5;
    while (i > 0) {
        printf("%d\n", i);
        i--;
    }
}
```




## While Loops (in 2D)

What's changing? What's always happening?

```
void a(void) {
    int row = 0;
    while (row < SIZE) {
        int col = 0;
        while (col < SIZE) {
            if (row == col) {
                printf("O");
            } else {
                printf("X");

            }
            col++;
        }
        row++;
        printf("\n");
    }
}
```

| row \ col | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| **0** | O | X | X | X |
| **1** | X | O | X | X |
| **2** | X | X | O | X |
| **3** | X | X | X | O |

## Structs and Enums

Structs!

```
#include <stdio.h>

struct person {
    int shoe_size;
    double height;
    char first_name_initial;
};
```

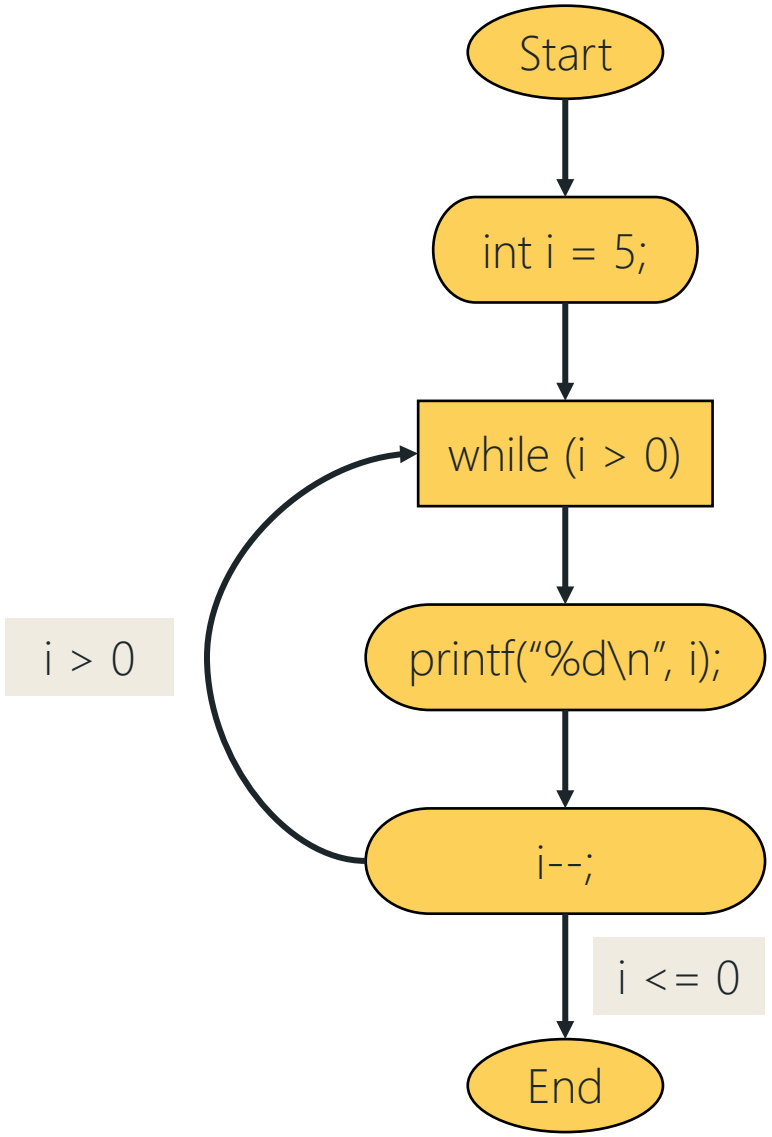
## Naming Conventions

Let's Kahoot!



# How does the code flow?

```
void a(void) {
    int i = 5;
    while (i > 0) {
        printf("%d\n", i);
        i--;
    }
}
```



# How does the code flow?

```
void b(void) {
    int i = 1;
    while (i < 32) {
        printf("%d\n", i);
        i = i + i;
    }
}
```

# How does the code flow?

```
void c(void) {
    int i = 0;
    while (i < 32) {
        printf("%d\n", i);
        i = i + 2;
    }
}
```

# How does the code flow?

```
void d(void) {
    int i = 5;
    while (i >= 0) {
        printf("%d\n", i);
        i--;
    }
}
```

# How does the code flow?

```
void e(void) {
    int i = 0;
    int keep_going = 1;
    while (keep_going == 1) {
        if (i > 3) {
            keep_going = 0;
        }
        i++;
    }
    printf("%d\n", i);
}
```

# How does the code flow?

```
void f(void) {
    int i;
    while (i > 0) {
        printf("%d\n", i);
        i--;
    }

}
```

# How does the code flow?

```
void g(void) {
    int i = 0;
    int max = 32;
    while (i < max) {
        printf("%d\n", i);
        max = max + 2;
    }

}
```

# How does the code flow?

```
void h(void) {
    int i = 0;
    int keep_going = 0;
    while (keep_going == 1) {
        if (i > 3) {
            keep_going = 0;
        }
        i++;
    }
    printf("%d\n", i);
}
```

# What's changing? What's always happening?

```
void a(void) {
    int row = 0;
    while (row < SIZE) {
        int col = 0;
        while (col < SIZE) {
            if (row == col) {
                printf("O");
            } else {
                printf("X");


            }
            col++;
        }
        row++;
        printf("\n");
    }
}
```

| row \ col | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| **0** | O | X | X | X |
| **1** | X | O | X | X |
| **2** | X | X | O | X |
| **3** | X | X | X | O |

# #define SIZE 4

```
void b(void) {
    int row = 0;
    while (row < SIZE) {
        int col = 0;
        while (col < SIZE) {
            if (col % 2 == 0) {
                printf("O");
            } else {
                printf("X");


            }
            col++;
        }
        row++;
        printf("\n");
    }
}
```

# #define SIZE 4

```
void c(void) {
    int row = 0;
    while (row < SIZE) {
        int col = 0;
        while (col < SIZE) {
            if (col != 1 && row != 1) {
                printf("O");
            } else {
                printf("X");


            }
            col++;
        }
        row++;
        printf("\n");
    }
}
```

# #define SIZE 4

```
void d(void) {
    int row = 0;
    while (row < SIZE) {
        printf("X");
        int col = 1;
        while (col < 3) {
            if (row == 0 || row == 3) {
                printf("X");
            } else {
                printf("O");
            }
            col++;
        }
        printf("X");
        row++;
        printf("\n");
    }
}
```

# Structs!

```c
#include <stdio.h>


struct person {
    int shoe_size;
    double height;
    char first_name_initial;
};
```

# Enums!

```c
#include <stdio.h>


enum opal_card_type {
    ADULT,
    STUDENT,
    CONCESSION
};
```

# Let’s Kahoot!


YOUR HONOUR
VARIABLES A1, A2,
AND A3 ARE COEFFICIENTS
imgflip.com


Descriptive variable name
Short variable name


int ab;
int rectangle_area;
imgflip.com

# Rules of good naming:

- Legality
  - Legal
    - Variables can contain
      - Letters
      - Numbers
      - _ (underscores)
  - Illegal
    - Variables cannot start with a number
- Stylistic
  - Good style
    - Variables start with a lowercase letter
    - `snake_case` is used
    - `#define` constants are in `SHOUTING_SNAKE_CASE`
  - Bad style
    - Variables are too short, or not descriptive
      - E.g. `feet`