COMP1911 22T2 (https://webcms3.cse.unsw.edu.au/COMP1911/22T2) Introduction to Programming (https://webcms3.cse.unsw.edu.au/COMP1911/22T2)

Code Examples from Lectures on 6-3_whileLoops

if.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/if.c)

A simple if statement that we will modify and turn into a while loop!

```
#include <stdio.h>

int main(void) {
    int i;
    i = 0;

    if(i < 10){
        printf("%d\n",i);
    }

    printf("Goodbye\n");
    return 0;
}
```

ifToWhile.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/ifToWhile.c)

An infinite while loop

Note: This is just for demonstration. We usually want to avoid infinite loops

Use Ctrl^C to terminate program

```
#include <stdio.h>

int main(void) {
    int i;

    i = 0;

    //We changed our if to a while
    //instead of doing the printf 0 or 1 time
    //it does it many times!
    // if(i < 10){
    while(i < 10){
        //because i is always 0 which is < 10 this loop is infinite
        //Use Ctrl^C to terminate program
        printf("%d\n",i);
    }
    printf("Goodbye\n");
    return 0;
}
```

whileForwards.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/whileForwards.c)

A simple while loop that prints numbers from 0..9

```
#include <stdio.h>

int main(void) {
    int i;

    i = 0;
    while(i < 10){
        printf("%d\n",i);
        i = i + 1;
    }

    printf("Goodbye\n");

    return 0;
}
```

whileReverse.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/whileReverse.c)

A simple while loop that prints numbers from 9..0

```
#include <stdio.h>

#include <stdio.h>

int main(void) {
    int i;

    i = 9;
    while(i >= 0){
        printf("%d\n");
        i = i - 1;
    }
    printf("\nGoodbye\n");

    return 0;
}
```

asterisks.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/asterisks.c)

A simple program demonstrating the use of scanf and while loops

```
#include <stdio.h>

#include <stdio.h>

int main(void) {
    int i;
    int numAsterisks;
    printf("How many asterisks? ");
    scanf("%d",&numAsterisks);

    i = 0;
    while(i < numAsterisks ){
        printf("*\n");
        i = i + 1;
    }
    printf("\nGoodbye\n");

    return 0;
}
```

whileFromTo.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/whileFromTo.c)

A simple while loop that prints numbers from min..max

And calculates the sum of numbers from min..max

```
#include <stdio.h>

int main(void) {
    int i;
    int max;
    int min;
    int sum;

    sum = 0;

    printf("Enter min :");
    scanf("%d",&min);
    printf("Enter max :");
    scanf("%d",&max);

    i = min;
    while(i <= max ){
        printf("%d\n",i);
        sum = sum + i;
        i = i + 1;
    }

    printf("Sum of %d..%d is %d\n",min,max,sum);

    return 0;
}
```

whileSentinel.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/whileSentinel.c)

A simple program demonstrating the use of scanf and reads user input until a user enters a -1

```
#include <stdio.h>

#define FALSE 0
#define TRUE 1
#define END_INPUT -1

int main(void) {
    int num;
    int stopLoop;

    printf("Enter numbers, Type %d to quit\n", END_INPUT);

    stopLoop = FALSE;
    while(stopLoop == FALSE){
        scanf("%d",&num);
        if(num == END_INPUT){
            stopLoop = TRUE;
        } else {
            printf("You entered %d\n",num);
        }
    }
    printf("\nGoodbye\n");

    return 0;
}
```

processInput.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/processInput.c)

A simple while loop that reads in and processes input

```
#include <stdio.h>

int main(void) {
    int i;
    int value;
    int howMany;
    double sum = 0;
    double average;

    printf("How many numbers? ");
    scanf("%d",&howMany);
    i = 0;
    while( i < howMany){
       scanf("%d",&value);
       sum = sum + value;

       i = i+1;
    }
    printf("Sum is  %.0lf\n",sum);
    average = sum/howMany;
    printf("Average is  %.2lf\n",average);
    return 0;
}
```

processInputSentinel.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/processInputSentinel.c)

A simple program demonstrating the use of scanf and reads user input until a user enters a -1 and calculates the sum and the average

```
#include <stdio.h>
#define FALSE 0
#define TRUE 1
#define END_INPUT -1

int main(void) {
    int num;
    int counter = 0;
    int stopLoop = FALSE;
    double sum = 0;
    double average;
    printf("Enter numbers, Type %d to quit\n", END_INPUT);

    while(stopLoop == FALSE){
        scanf("%d",&num);
        if(num == END_INPUT){
            stopLoop = TRUE;
        } else {
            printf("You entered %d\n",num);
            sum = sum + num;
            counter = counter + 1;
        }
    }
    if(counter > 0){
        printf("The sum was %lf\n",sum);
        printf("The average was %lf\n",sum/counter);
        printf("\nGoodbye\n");
    } else {
        printf("You entered no data\n");
    }
    return 0;
}
```

max0.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/max0.c)

A simple while loop that reads in integers and finds the maximum

```
#include <stdio.h>
#include <limits.h>

int main(void) {
    int i;
    int value;
    //initiliase max to smallest possible value
    int max = INT_MIN;
    int howMany;

    printf("How many numbers? ");
    scanf("%d",&howMany);

    i = 0;
    while( i < howMany){
       scanf("%d",&value);
       if(value > max){
          max = value;
       }

       i = i+1;
    }
    if(howMany > 0){
        printf("The max is %d\n",max);
    }
    return 0;
}
```

max1.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/max1.c)

A simple while loop that reads in integers and finds the maximum

```
#include <stdio.h>

int main(void) {
    int i;
    int value;
    //instead of initiliasing max to smallest possible value,
    //initialise it to the first value read in during the loop
    int max;
    int howMany;

    printf("How many numbers? ");
    scanf("%d",&howMany);

    i = 0;
    while( i < howMany){
       scanf("%d",&value);
       if(i == 0){
          max = value;
       } else if(value > max){
          max = value;
       }

       i = i+1;
    }
    if(howMany > 0){
        printf("The max is %d\n",max);
    }
    return 0;
}
```

printEven0.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/printEven0.c)

Exercise: print even numbers 1 <= n

```
#include <stdio.h>



int main(void) {
    int i;
    int max;

    printf("Enter max :");
    scanf("%d",&max);

    i = 1;
    while(i <= max ){
        if(i % 2 == 0){
            printf("%d is even\n",i);
        }
        i = i + 1;
    }

    printf("Goodbye\n");

    return 0;
}
```

printEven1.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/printEven1.c)

Exercise: print even numbers 1 <= n

```
#include <stdio.h>



int main(void) {
    int i;

    int max;


    printf("Enter max :");
    scanf("%d",&max);

    //Only generate even numbers
    i = 2;
    while(i <= max ){
        printf("%d is even\n",i);

        i = i + 2;
    }

    printf("Goodbye\n");

    return 0;
}
```

factorial.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/factorial.c)

Calculating factorial

Note: factorial grows very quickly so int, long long, unsigned long long and double are not sufficient and will overflow and/or produce incorrect results for even not so large values of n

```
#include <stdio.h>


int main(void){

    // n! = 1*2*3..*n
    int i;
    int n;
    double factorial = 1;

    i = 1;

    printf("Enter a value of n <= 20 ");
    scanf("%d",&n);

    while( i <= n) {
        factorial = factorial * i;
        printf("%d %.0lf\n",i, factorial);
        i = i+1;
    }

    printf("%d! = %.0lf\n",n,factorial);

    return 0;
}
```

thirteen_a.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/thirteen_a.c)

A silly program which prints first 1000 multiples of 13

```
#include <stdio.h>

int main(void) {
    printf("%d\n", 13);
    printf("%d\n", 26);
    printf("%d\n", 39);
    printf("%d\n", 52);
    printf("%d\n", 65);
    printf("%d\n", 78);
    printf("%d\n", 91);
    printf("%d\n", 104);
    printf("%d\n", 117);
    printf("%d\n", 130);
    printf("%d\n", 143);
    printf("%d\n", 156);
    printf("%d\n", 169);
    printf("%d\n", 182);
    printf("%d\n", 195);
    printf("%d\n", 208);
    printf("%d\n", 221);
    printf("%d\n", 234);
    printf("%d\n", 247);
    printf("%d\n", 260);
    printf("%d\n", 273);
    printf("%d\n", 286);
    printf("%d\n", 299);
    printf("%d\n", 312);
    printf("%d\n", 325);
    printf("%d\n", 338);
    printf("%d\n", 351);
    printf("%d\n", 364);
    printf("%d\n", 377);
...
```

thirteen_b.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/thirteen_b.c)

A simple program which uses a while loop to print first 1000 multiples of 13

```
#include <stdio.h>

int main(void) {
    int i;
    i = 1;
    while (i <= 13000) {
        if (i % 13 == 0) {
            printf("%d\n", i);
        }
        i = i + 1;
    }

    return 0;
}
```

thirteen_c.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/thirteen_c.c)

Another simple program which uses a while loop to print first 1000 multiples of 13

```
#include <stdio.h>

int main(void) {
    int i;
    i = 13;
    while (i <= 13000) {
        printf("%d\n", i);
        i = i + 13;
    }
    return 0;
}
```

thirteen_d.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/thirteen_d.c)

Another simple program which uses a while loop to print first 1000 multiples of 13

```
#include <stdio.h>
#define NUM_MULTIPLES 1000

int main(void) {
    int i;
    i = 1;
    while (i <= NUM_MULTIPLES) {
        printf("%d\n", i*13);
        i = i + 1;
    }
    return 0;
}
```

thirteen_e.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/thirteen_e.c)

Another simple program which uses a while loop to print first 1000 multiples of 13

```
#include <stdio.h>
#define NUM_MULTIPLES 1000

int main(void) {
    int i;
    int value = 13;
    i = 0;
    while (i < NUM_MULTIPLES ) {
        printf("%d\n", value);
            value = value + 13;
        i = i + 1;
    }
    return 0;
}
```

series.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/series.c)

Sum the series 1/1 + 1/2 + 1/3 + 1/4 + ...

```
#include <stdio.h>


#define N_SERIES_TERMS 1000

int main(void) {
    int i;
    double sum = 0;
    i = 1;
    while(i <= N_SERIES_TERMS){
        printf("%d\n",i);
        sum = sum + 1.0/i;
        i = i + 1;
    }

    printf("Sum from 1 to %d is %lf\n",N_SERIES_TERMS,sum);
    return 0;
}
```

converge.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/converge.c)

Calculate the the series 1/1 + 1/2 + 1/4 + 1/8 + ... until it converges to within 10 decimal places

```
#include <stdio.h>

#define MIN 1.0e-10

int main(void) {
    long long i = 1;
    double sum = 0;
    double term = 1;

    while(term > MIN ){
       term = 1.0/i;
       sum = sum + term;
       printf("%lld %lf %.15lf\n",i,sum,term);
       i = i * 2;
    }

    printf("1 + 1/2 + 1/4 + 1/8 + ...  = %.10f\n", sum);
    return 0;
}
```

calculate_e.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/calculate_e.c)

Calculate the mathematical constant e by summing the series 1 + 1/1 + 1/(1*2) + 1/(1*2*3) + 1/(1*2*3*4) + ...
http://en.wikipedia.org/wiki/E_%28mathematical_constant%29 (http://en.wikipedia.org/wiki/E_%28mathematical_constant%29)
2.71828

```
#include <stdio.h>

#define N_SERIES_TERMS 20

int main(void) {
    double factorial = 1;
    double e = 1;
    int i = 1;

    while(i <= N_SERIES_TERMS){
        factorial = factorial * i;
        e = e + 1/factorial;
        //printf("%d %lf %lf\n",i, factorial,inverse);
        i = i + 1;
    }
    printf("e is %lf\n",e);
    return 0;
}
```

nestedWhile0.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/nestedWhile0.c)

A simple nested while loop

```
#include <stdio.h>


int main(void) {
    int i;
    int j;

    i = 0;
    while(i < 5 ){
       j = 0;
       while(j < 10 ){
           printf("%d ",j);

           j = j + 1;
       }
       printf("\n");
       i = i + 1;
    }
    printf("\nGoodbye\n");

    return 0;
}
```

nestedWhile1.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/nestedWhile1.c)

A simple nested while loop

```
#include <stdio.h>


int main(void) {
    int i;
    int j;

    i = 0;
    while(i < 5 ){
       j = 0;
       while(j < 5 ){
           printf("(%d,%d) ",i,j);

           j = j + 1;
       }
       printf("\n");
       i = i + 1;
    }
    printf("\nGoodbye\n");

    return 0;
}
```

rect.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/rect.c)

A simple nested while loop printing rectangle of *

```
#include <stdio.h>

#define ROWS 10
#define COLS 5

int main(void) {
    int i;
    int j;

    i = 0;
    while(i < ROWS ){
       j = 0;
       while(j < COLS ){
           printf("*");
           j = j + 1;
       }
       printf("\n");
       i = i + 1;
    }
    printf("\nGoodbye\n");

    return 0;
}
```

stripe.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/stripe.c)

A simple nested while loop printing rectangle of *

With a stripe along the middle row

```
#include <stdio.h>

#define ROWS 5
#define COLS 5

int main(void) {
    int i;
    int j;

    i = 0;
    while(i < ROWS ){
       j = 0;
       while(j < COLS ){
           if(i == ROWS/2){
               printf("!");
           } else {
               printf("*");
           }
           j = j + 1;
       }
       printf("\n");
       i = i + 1;
    }
    printf("\nGoodbye\n");

    return 0;
}
```

box.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/box.c)

A simple while loop printing a box

```
#include <stdio.h>

#define HEIGHT 5
#define WIDTH 5

int main(void) {
    int i;
    int j;

    i = 0;
    while(i < HEIGHT ){
       j = 0;
       while(j < WIDTH ){

           if(i == 0 || i == HEIGHT -1  ||
              j == 0 || j == WIDTH - 1){
               printf("*");
           } else {
               printf(" ");
           }
           j = j + 1;
       }
       printf("\n");
       i = i + 1;
    }


    return 0;
}
```

starsAndStripes.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/starsAndStripes.c)

A simple program demonstrating nested while loops

```
#include <stdio.h>


int main(void) {
    int col;
    int row;
    int width;
    int height;
    printf("How wide? ");
    scanf("%d",&width);
    printf("How high? ");
    scanf("%d",&height);

    row = 0;
    while(row < height){
        col = 0;
        while(col < width ){
            if( col % 2 == 0){
               printf("*");
            } else {
               printf("|");
            }
            col = col + 1;
        }
        printf("\n");
        row = row + 1;
    }
    printf("\nGoodbye\n");

    return 0;
}
```

cross.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/cross.c)

A simple nested while loop printing an X

```
#include <stdio.h>

#define SIZE 5


int main(void) {
    int i;
    int j;

    i = 0;
    while(i < SIZE ){
       j = 0;
       while(j < SIZE ){
           if(i == j || i+j == SIZE -1 ){
               printf("*");
           } else {
               printf(" ");
           }
           j = j + 1;
       }
       printf("\n");
       i = i + 1;
    }


    return 0;
}
```

processInputNested.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/processInputNested.c)

A simple while loop that reads in and processes input reads in numbers from user

```
#include <stdio.h>
#define FALSE 0
#define TRUE 1


int main(void) {
    double sum = 0;
    double value;
    int howMany;
    int i;
    int stopLoop = FALSE;

    while(stopLoop == FALSE ){
        printf("How many numbers will you enter? ");
        scanf("%d",&howMany);
        if ( howMany > 0 ) {
            printf("Enter your %d numbers\n",howMany);
            sum = 0;
            i = 0;
            while(i < howMany){
                scanf("%lf",&value);
                sum = sum + value;
                i = i + 1;
            }

            printf("Sum is  %lf\n",sum);
        } else {
            stopLoop = TRUE;
        }
    }
    return 0;
}
```

pythag.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/pythag.c)

A simple program which searches for pythagorean triples integers which form form the sides of a right triangle http://en.wikipedia.org/wiki/Pythagorean_triple (http://en.wikipedia.org/wiki/Pythagorean_triple)

Andrew Taylor

```
#include <stdio.h>

//a^2 + b^2 = c^2
#define N 100

int main(void){
    int x;
    int y;
    int z;

    x = 1;
    while(x <= N){
        y = 1;
        while( y <= N){
            z = 1;
            while( z <= N){
                //condition x < y so we don't get
                //different permutations of the
                //same three sides eg
                //3 4 5
                //4 3 5
                if(x*x + y*y == z*z && x < y){
                    printf("%d %d %d\n",x,y,z);
                }
                z = z + 1;
            }
            y = y + 1;
        }
        x = x + 1;
    }
    return 0;
}
```

exerciseCalcPi.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/exerciseCalcPi.c)

Calculate the mathematical constant pi by summing the series 4 - 4/3 + 4/5 - 4/7 + 4/9 + ... with 1000000 terms

```
#include <stdio.h>

//This has been left as an exercise and we will answer it next week

#define N_SERIES_TERMS 1000000


int main(void) {
    double sum = 0;


    printf("4 - 4/3 + 4/5 - 4/7 + 4/9 + ... = %lf\n", sum);
    return 0;
}
```

calculatePi.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/calculatePi.c)

Calculate the mathematical constant pi to 6 decimal places by summing the series 4 - 4/3 + 4/5 - 4/7 + 4/9 + ...

This is the solution we developed in class for the exercise

```
#include <stdio.h>


#define N_SERIES_TERMS 1000000


int main(void) {
    double sum = 0;
    int denominator = 1;
    int counter = 1;
    while(counter <= N_SERIES_TERMS){
        if(counter % 2 == 1){
           sum = sum + 4.0/denominator;
        } else {
           sum = sum - 4.0/denominator;
        }
        denominator = denominator + 2;
        counter = counter + 1;
    }

    printf("4 - 4/3 + 4/5 - 4/7 + 4/9 + ... = %lf\n", sum);
    return 0;
}
```

snap.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/snap.c)

A simple program which reads integers and prints snap and exits if the same number is read twice in a row

Note for simplicity we are assuming scanf succeeds in reading an integer.

```
#include <stdio.h>
#define FALSE 0
#define TRUE 1

// 1 5 6 6
// p p p c
int main(void) {
    int previousCard;
    int card;
    int snap = FALSE;
    scanf("%d",&previousCard);
    scanf("%d",&card);

    while(snap == FALSE){
        if(previousCard == card){
            snap = TRUE;
        } else {
            previousCard = card;
            scanf("%d",&card);
        }
    }

    printf("Snap!\n");
    return 0;
}
```

fib.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/fib.c)

Print first 20 fibonacci numbers

```
#include <stdio.h>

#define HOWMANY 20
// 0 1  1 2 3 5 8 13 21 34
// p p  p p c
int main(void) {
    int previous = 0;
    int current = 1;
    int fib;

    int counter = 1;

    while(counter <= HOWMANY){
        fib = previous + current;
        printf("%d\n",fib);
        previous = current;
        current = fib;
        counter = counter + 1;
    }

    return 0;
}
```

lec_pool.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/lec_pool.c)

```
#include <stdio.h>

#define POOL_SIZE 1000

int main(void) {
    double poolSize = POOL_SIZE;
    int daysToDrain;
    int i = 0;
    scanf("%d", &daysToDrain);
    while (i < daysToDrain) {
        poolSize = poolSize * 0.7;
        i = i + 1;
    }
    printf("Size after %d days is %lfL\n", daysToDrain, poolSize);
    return 0;
}
```

lec_five.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/lec_five.c)

```
#include <stdio.h>

int main(void) {
    int n;
    int i;
    scanf("%d", &n);
    i = n;
    while (i <= n + 10) {
        printf("%d\n", i);
        i = i + 1;
    }
    return 0;
}
```

lec_round.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/lec_round.c)

```
#include <stdio.h>

int main(void) {

    int a, b, c;
    double total;
    double denominator = 2;
    a = 3;
    b = 7;
    c = 11;

    total = (a + b + c) / denominator;
    printf("%lf\n", total);
    return 0;
}

// double / double => double
// double / int => double
// int / double => double
// int / int => int
```

lec_sent.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/lec_sent.c)

```
#include <stdio.h>

#define FALSE 0
#define TRUE 1

int main(void) {
    int keepLooping, number;
    keepLooping = TRUE;
    while (keepLooping == TRUE) {
        scanf("%d", &number);
        if (number == 0) {
            keepLooping = FALSE;
        } else {
            printf("%d\n", number);
        }
    }
    return 0;
}
```

lec_randommath.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/lec_randommath.c)

```
#include <stdio.h>
#include <math.h>

int main(void) {
    double a, b, c, x, discriminant;
    printf("Give me a: \n");
    scanf("%lf", &a);
    printf("Give me b: \n");
    scanf("%lf", &b);
    printf("Give me c: \n");
    scanf("%lf", &c);
    discriminant = sqrt(b*b-4*a*c);
    x = (-b + discriminant) / (2 * a);
    printf("%lf\n", x);
    return 0;
}
```

lec_down.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/lec_down.c)

```
#include <stdio.h>

int main(void) {
    int number;
    int i = 0;
    scanf("%d", &number);
    while (i <= 50 - number) {
        printf("%d\n", 50 - i);
        i = i + 1;
    }
    return 0;
}
```

lec_stuff.c (https://cgi.cse.unsw.edu.au/~cs1911/22T2/lec/6-3_whileLoops/code/lec_stuff.c)

```
#include <stdio.h>

int main(void) {
    int x = 4294967295;
    unsigned int y = 4294967295;

    printf("%b\n", x);
    printf("%b\n", y);
    return 0;
}
```