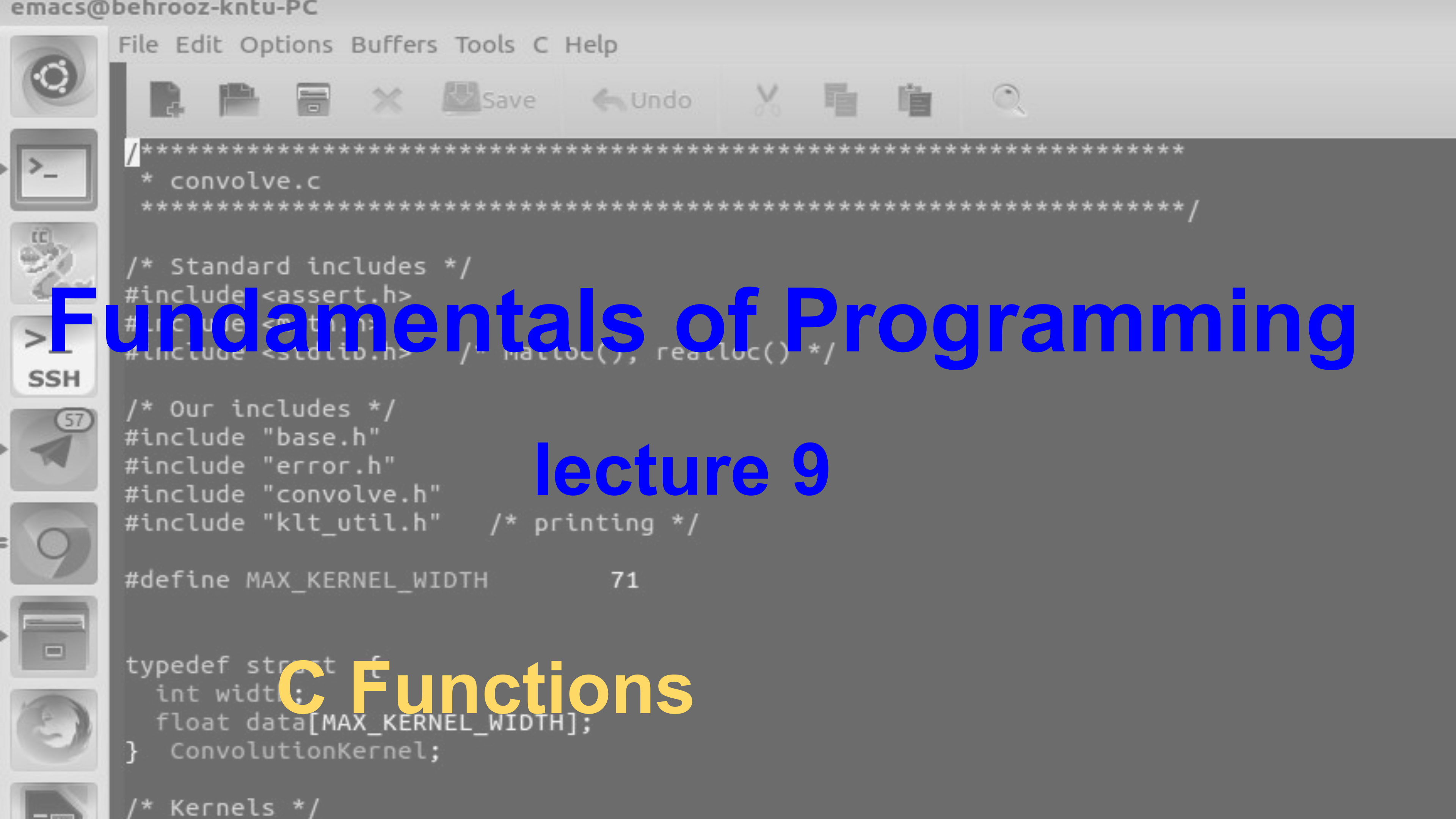
Fundamentals of Programming

lecture 9

C Functions



```
/*********************************************************************
 * convolve.c
 *********************************************************************/

/* Standard includes */
#include <assert.h>
#include <math.h>
#include <stdlib.h>   /* malloc(), realloc() */

/* Our includes */
#include "base.h"
#include "error.h"
#include "convolve.h"
#include "klt_util.h"   /* printing */

#define MAX_KERNEL_WIDTH        71


typedef struct  {
  int width;
  float data[MAX_KERNEL_WIDTH];
}  ConvolutionKernel;

/* Kernels */
```

What about more complicated math?

+ − * / %

What about more complicated math?

```
3 * x = 2
```

```
x*x = 2
```

$3^x = 4$

$e^x = 2$

```
sin(x) = .6
```

What about more complicated math?

$$x*x = 2$$

$$3^x = 4$$

$$e^x = 2$$

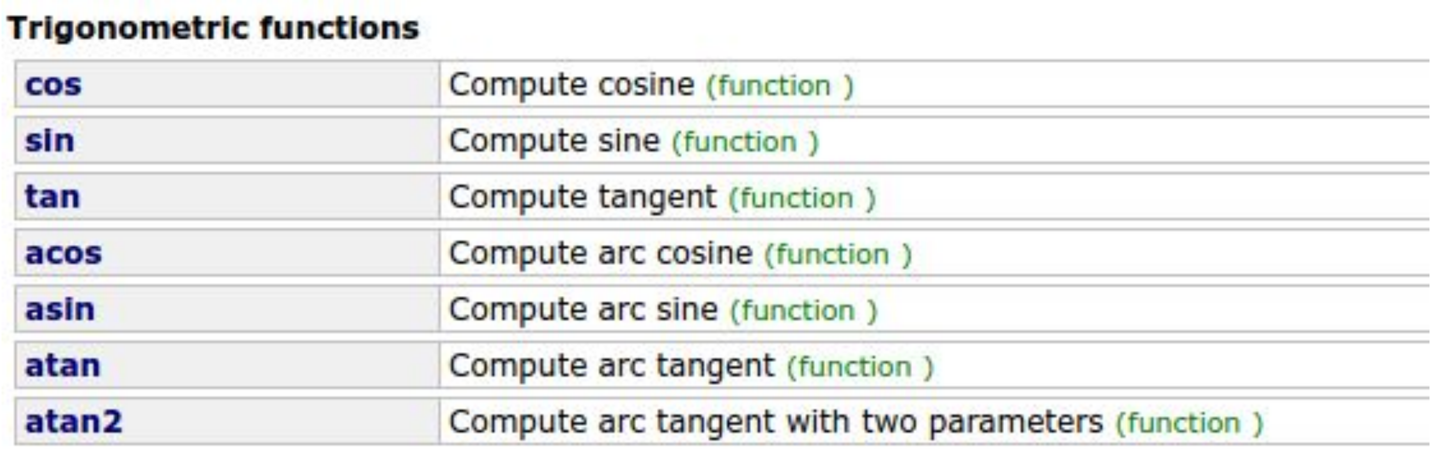
$$\sin(x) = .6$$

$$\arctan(x) = 10$$

Standard C library

- ANSI C standard (ISO C)
    - C89, C90, C95, C99, C11, **C18**
    - **gcc -std=c90**
- **C Standard Library:** functions, types, macros:
    - Math
    - I/O
    - string operations
    - memory management
    - system
- https://www.slideshare.net/teach4uin/stdlib-functions-lesson
- https://www.tutorialspoint.com/c_standard_library

Math

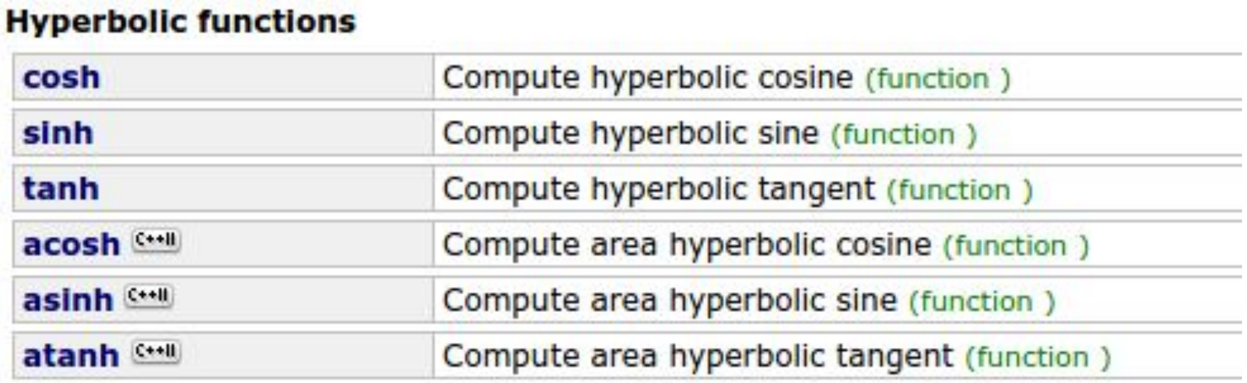


Trigonometric functions

cos	Compute cosine (function)
sin	Compute sine (function)
tan	Compute tangent (function)
acos	Compute arc cosine (function)
asin	Compute arc sine (function)
atan	Compute arc tangent (function)
atan2	Compute arc tangent with two parameters (function)

http://www.cplusplus.com/reference/cmath/

Math

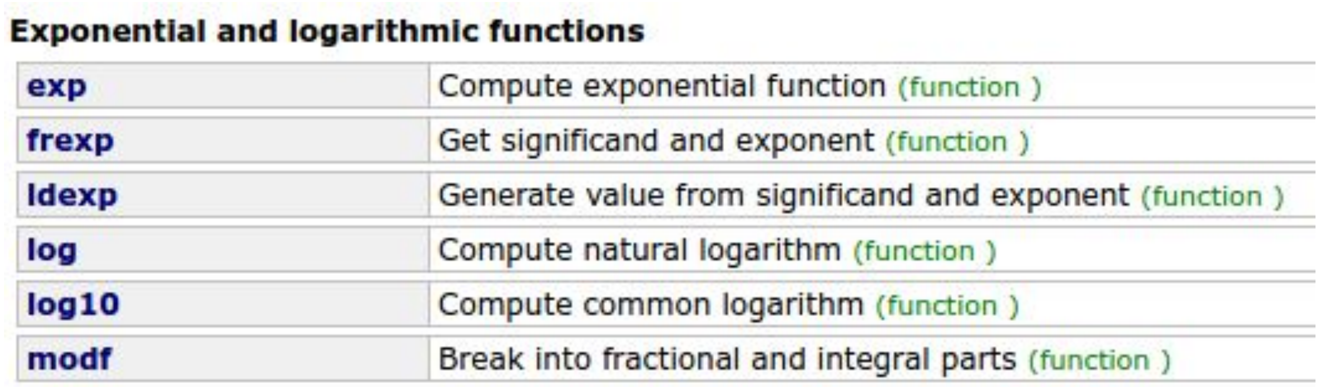


Hyperbolic functions

cosh	Compute hyperbolic cosine (function)
sinh	Compute hyperbolic sine (function)
tanh	Compute hyperbolic tangent (function)
acosh C++11	Compute area hyperbolic cosine (function)
asinh C++11	Compute area hyperbolic sine (function)
atanh C++11	Compute area hyperbolic tangent (function)

http://www.cplusplus.com/reference/cmath/

Math

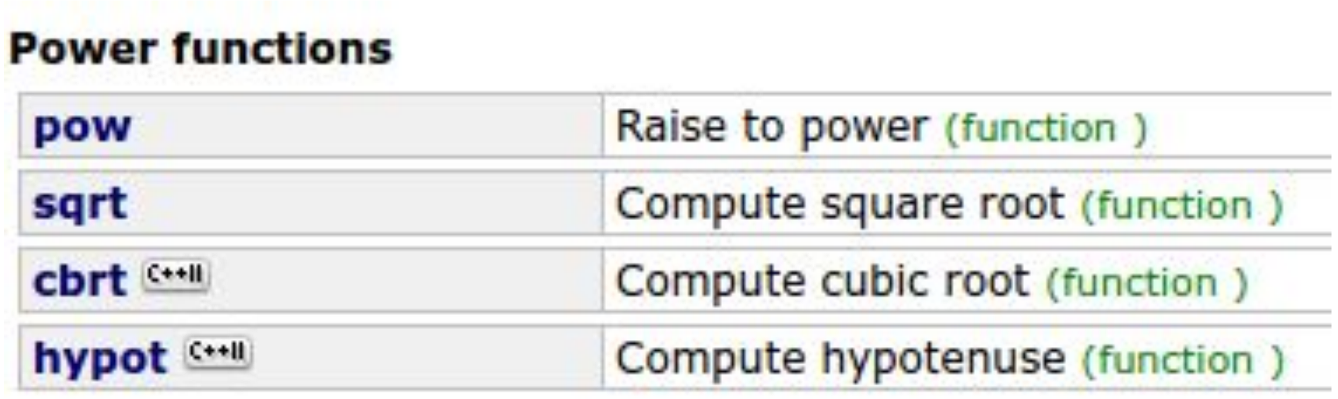


Exponential and logarithmic functions

exp	Compute exponential function (function)
frexp	Get significand and exponent (function)
ldexp	Generate value from significand and exponent (function)
log	Compute natural logarithm (function)
log10	Compute common logarithm (function)
modf	Break into fractional and integral parts (function)

http://www.cplusplus.com/reference/cmath/

Math

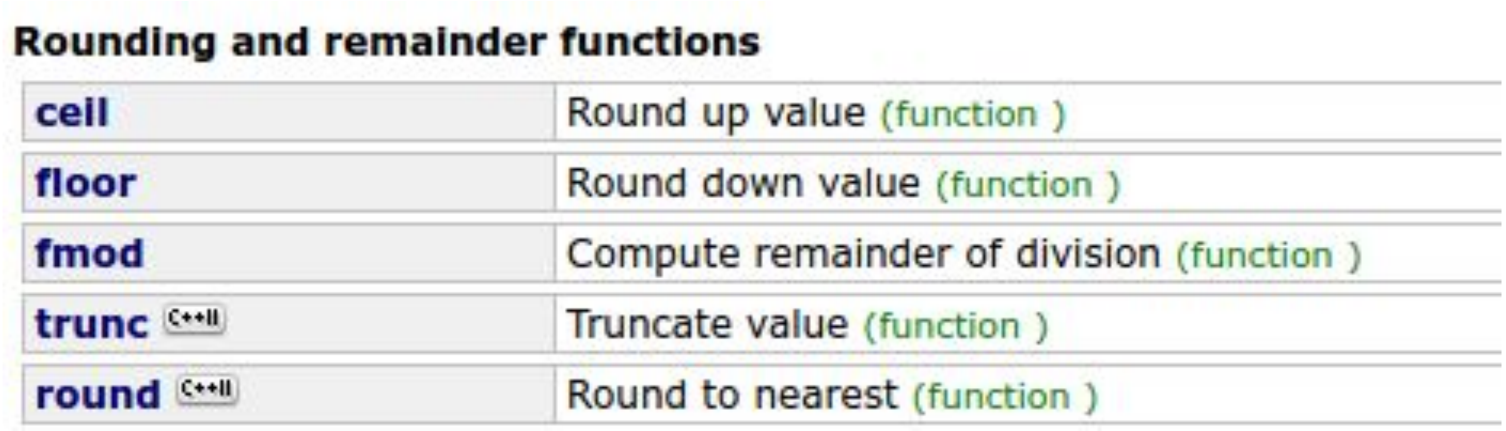


Power functions

pow	Raise to power (function)
sqrt	Compute square root (function)
cbrt C++11	Compute cubic root (function)
hypot C++11	Compute hypotenuse (function)

http://www.cplusplus.com/reference/cmath/

Math

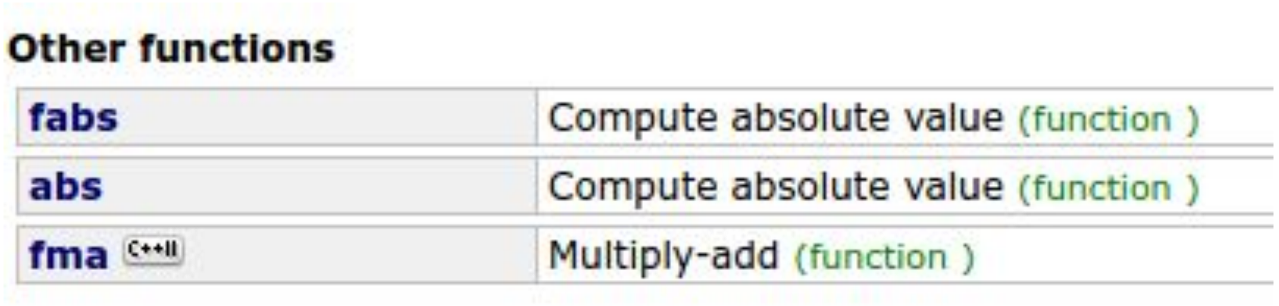


Rounding and remainder functions

ceil	Round up value (function)
floor	Round down value (function)
fmod	Compute remainder of division (function)
trunc C++11	Truncate value (function)
round C++11	Round to nearest (function)

http://www.cplusplus.com/reference/cmath/

Math



Other functions

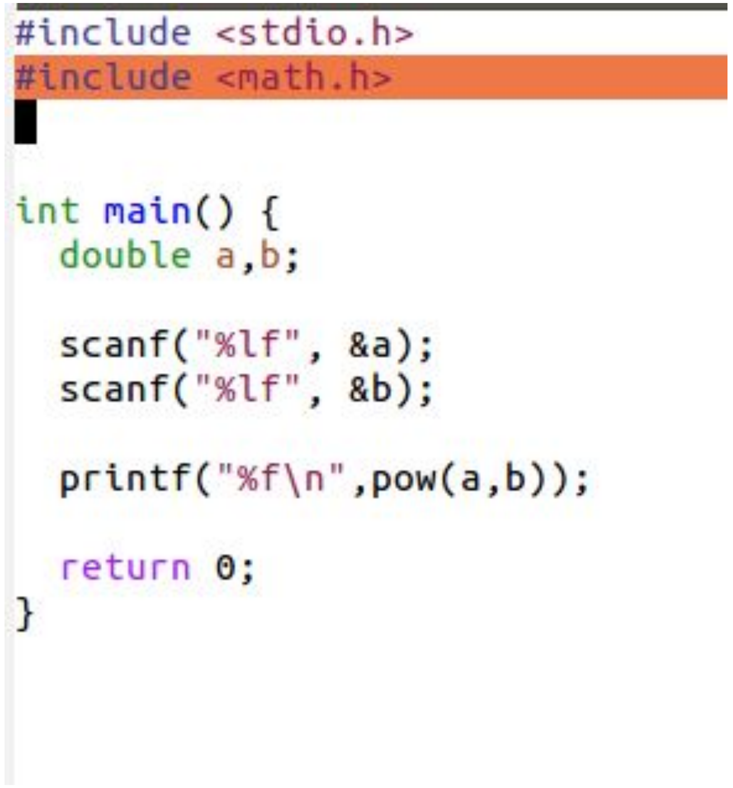
fabs	Compute absolute value (function)
abs	Compute absolute value (function)
fma C++11	Multiply-add (function)

http://www.cplusplus.com/reference/cmath/

Math

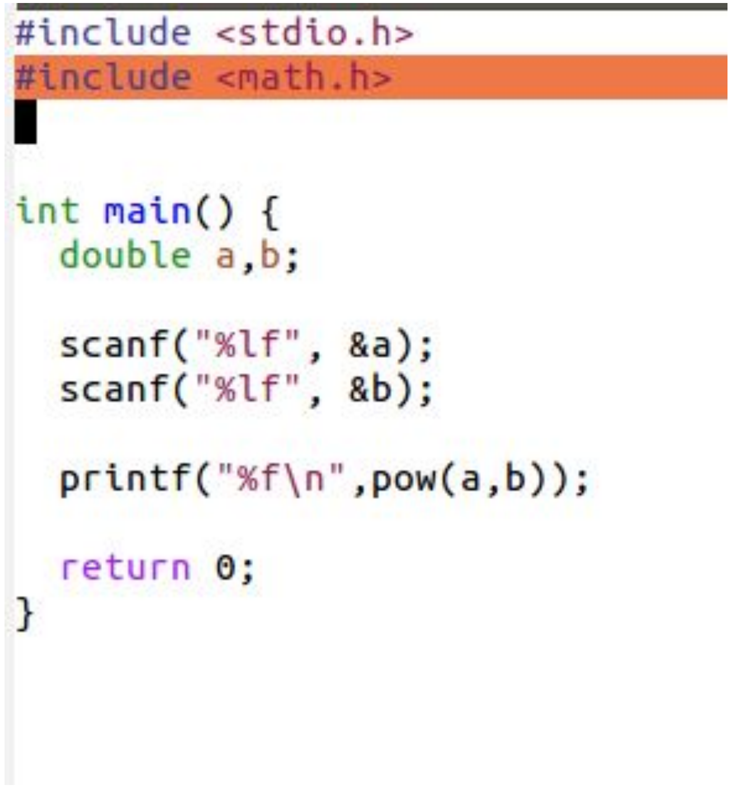
```
#include <math.h>
```

Math



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


int main() {
  double a,b;

  scanf("%lf", &a);
  scanf("%lf", &b);

  printf("%f\n",pow(a,b));

  return 0;
}
```

Math



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


int main() {
  double a,b;

  scanf("%lf", &a);
  scanf("%lf", &b);

  printf("%f\n",pow(a,b));

  return 0;
}
```

Compile:

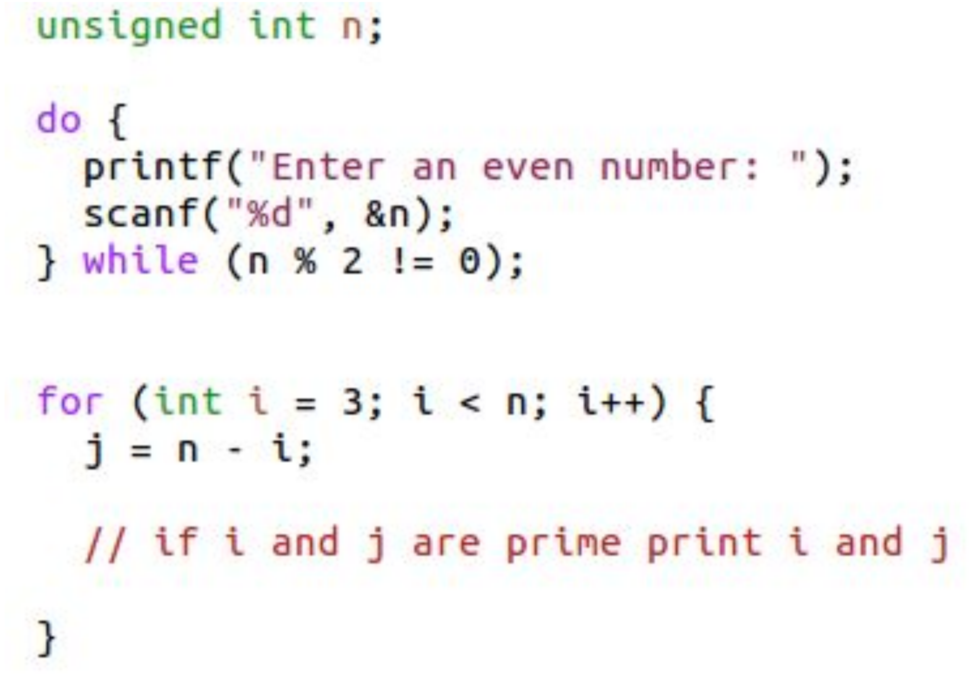
gcc testpow.c -l m

Why functions?

Goldbach's conjecture

every even integer (>2) is sum of two primes

Goldbach's conjecture



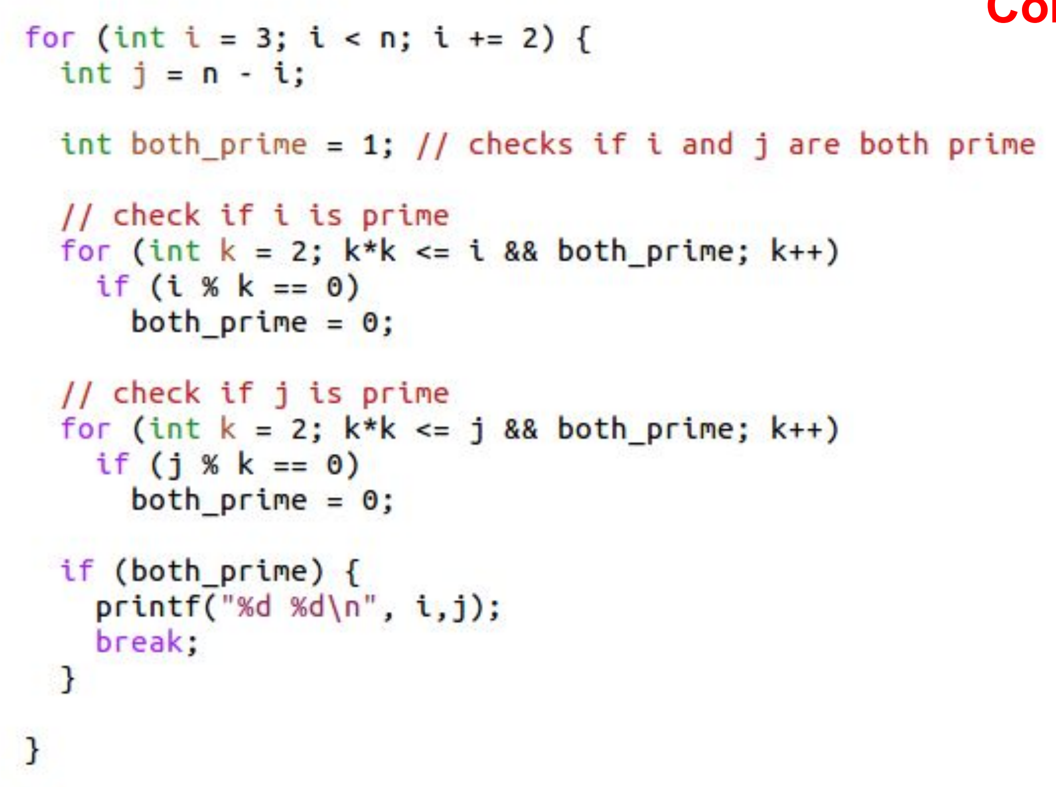
```
unsigned int n;

do {
  printf("Enter an even number: ");
  scanf("%d", &n);
} while (n % 2 != 0);


for (int i = 3; i < n; i++) {
  j = n - i;

  // if i and j are prime print i and j

}
```

Goldbach's conjecture

```c
for (int i = 3; i < n; i += 2) {
  int j = n - i;

  int both_prime = 1; // checks if i and j are both prime

  // check if i is prime
  for (int k = 2; k*k <= i && both_prime; k++)
    if (i % k == 0)
      both_prime = 0;

  // check if j is prime
  for (int k = 2; k*k <= j && both_prime; k++)
    if (j % k == 0)
      both_prime = 0;

  if (both_prime) {
    printf("%d %d\n", i,j);
    break;
  }

}
```

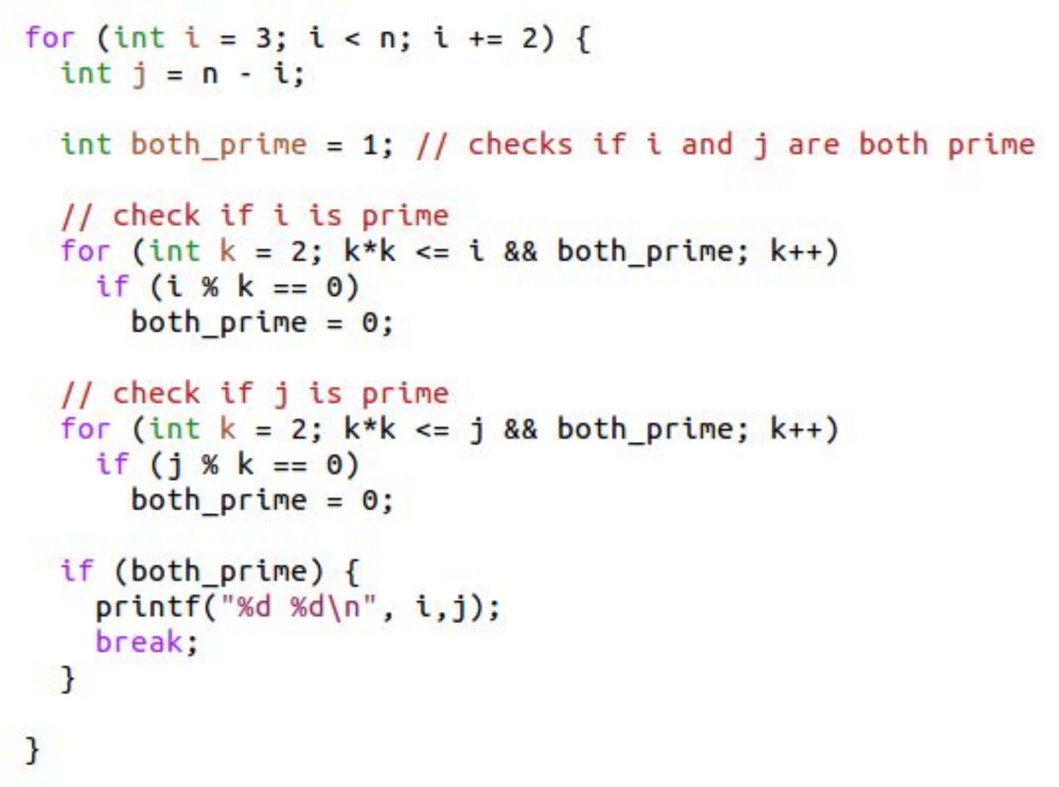
Goldbach's conjecture

Comments!



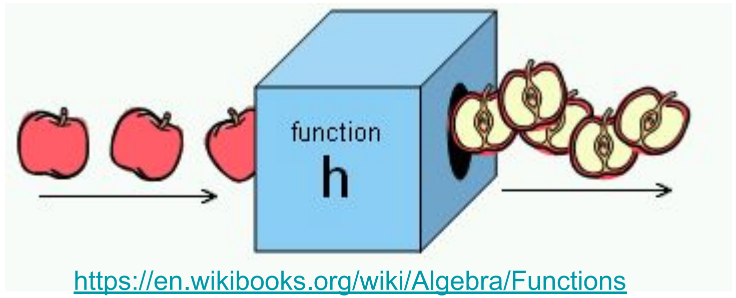
```
for (int i = 3; i < n; i += 2) {
  int j = n - i;

  int both_prime = 1; // checks if i and j are both prime

  // check if i is prime
  for (int k = 2; k*k <= i && both_prime; k++)
    if (i % k == 0)
      both_prime = 0;

  // check if j is prime
  for (int k = 2; k*k <= j && both_prime; k++)
    if (j % k == 0)
      both_prime = 0;

  if (both_prime) {
    printf("%d %d\n", i,j);
    break;
  }

}
```

What's wrong?

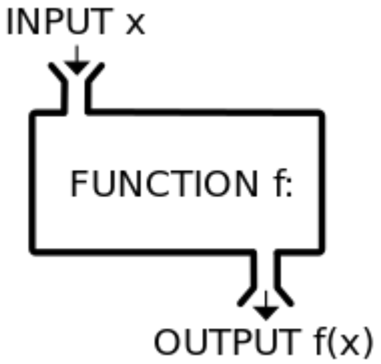


```
for (int i = 3; i < n; i += 2) {
  int j = n - i;

  int both_prime = 1; // checks if i and j are both prime

  // check if i is prime
  for (int k = 2; k*k <= i && both_prime; k++)
    if (i % k == 0)
      both_prime = 0;

  // check if j is prime
  for (int k = 2; k*k <= j && both_prime; k++)
    if (j % k == 0)
      both_prime = 0;

  if (both_prime) {
    printf("%d %d\n", i,j);
    break;
  }

}
```

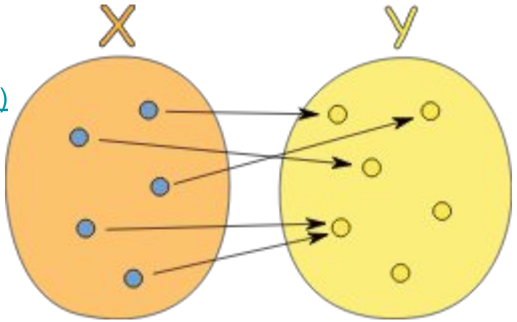
Functions

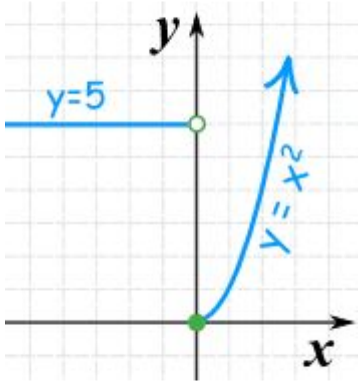


https://en.wikibooks.org/wiki/Algebra/Functions



https://en.wikipedia.org/wiki/Function_(mathematics)

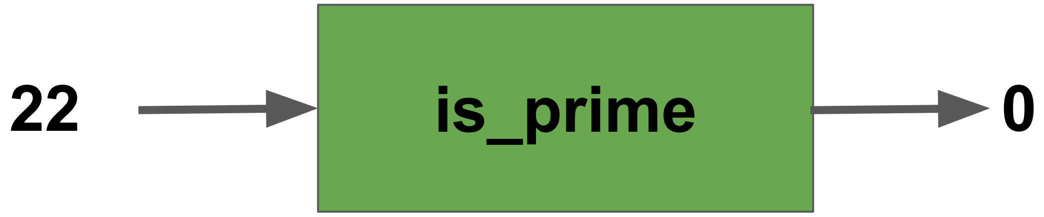




https://www.mathsisfun.com/sets/function.html

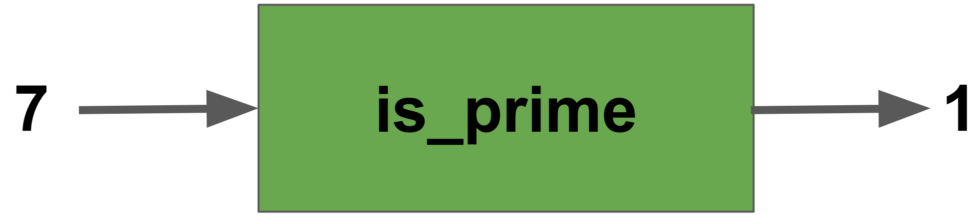
A function for detecting prime numbers

Write a function which gets a number and returns 1 if it is prime and 0 otherwise



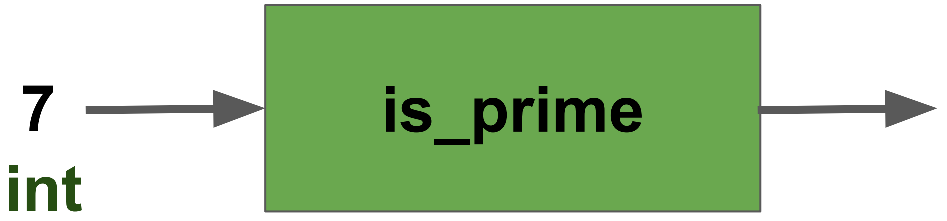
A function for detecting prime numbers

Write a function which gets a number and returns 1 if it is prime and 0 otherwise



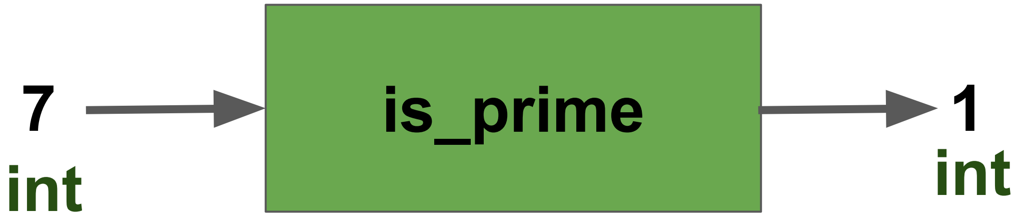
A function for detecting prime numbers

Write a function which gets a number and returns 1 if it is prime and 0 otherwise

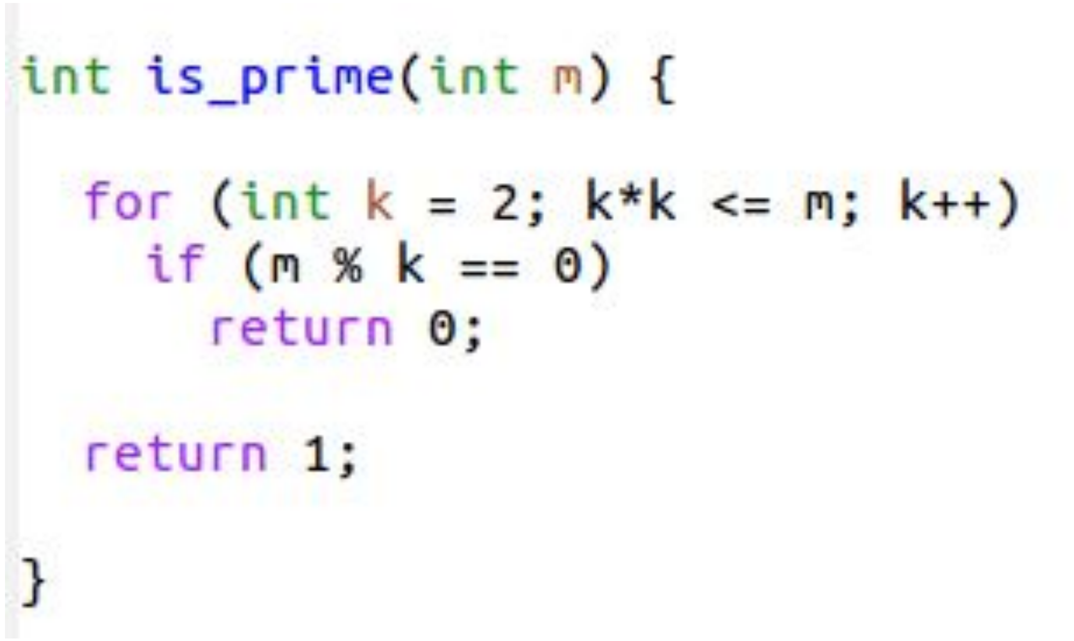


A function for detecting prime numbers

Write a function which gets a number and returns 1 if it is prime and 0 otherwise

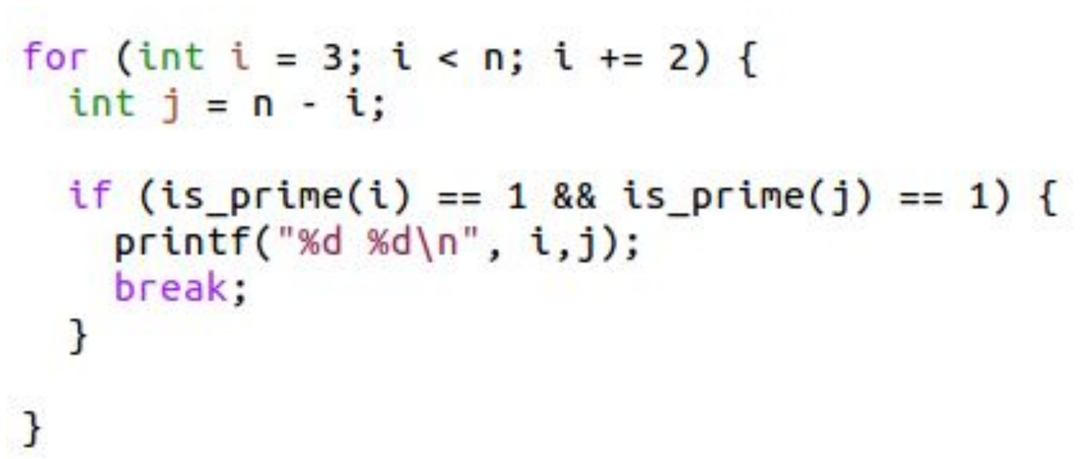


A function for detecting prime numbers

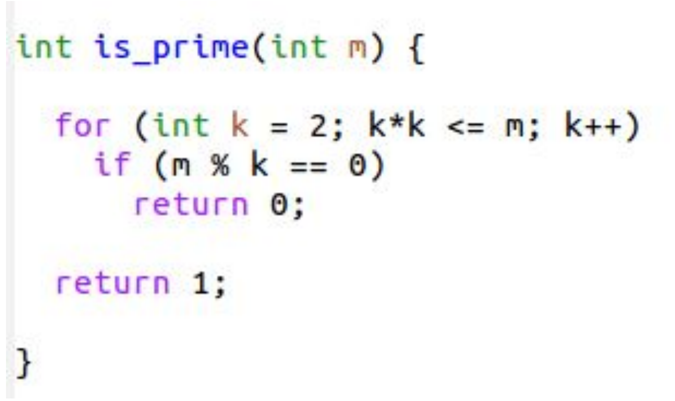


```c
int is_prime(int m) {

  for (int k = 2; k*k <= m; k++)
    if (m % k == 0)
      return 0;

  return 1;

}
```

A function for detecting prime numbers

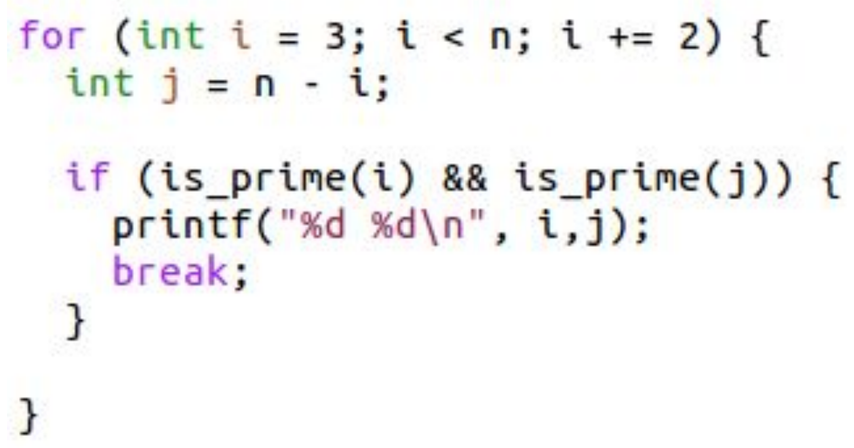


```c
for (int i = 3; i < n; i += 2) {
  int j = n - i;

  if (is_prime(i) == 1 && is_prime(j) == 1) {
    printf("%d %d\n", i,j);
    break;
  }

}
```

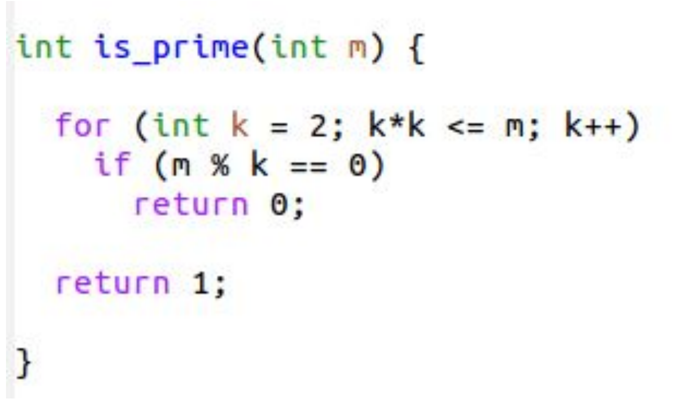


```c
int is_prime(int m) {

  for (int k = 2; k*k <= m; k++)
    if (m % k == 0)
      return 0;

  return 1;

}
```

A function for detecting prime numbers

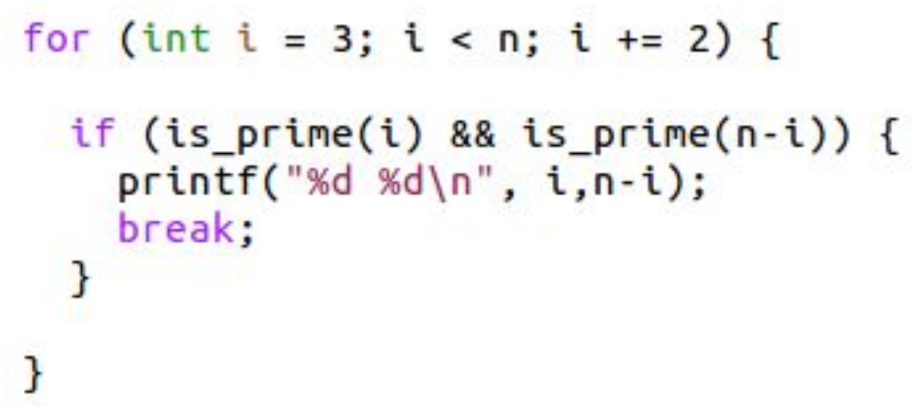


```c
for (int i = 3; i < n; i += 2) {
  int j = n - i;

  if (is_prime(i) && is_prime(j)) {
    printf("%d %d\n", i,j);
    break;
  }

}
```

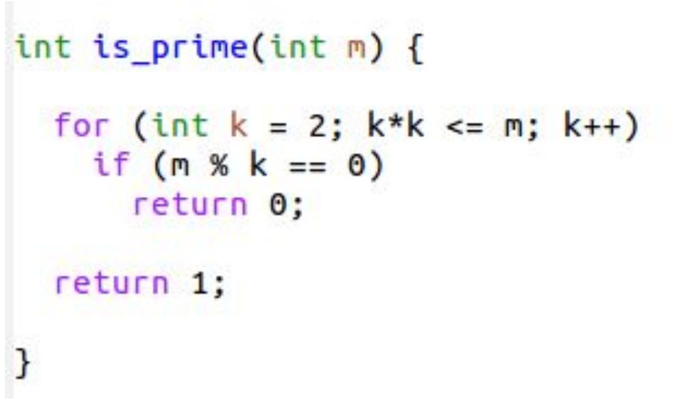


```c
int is_prime(int m) {

  for (int k = 2; k*k <= m; k++)
    if (m % k == 0)
      return 0;

  return 1;

}
```

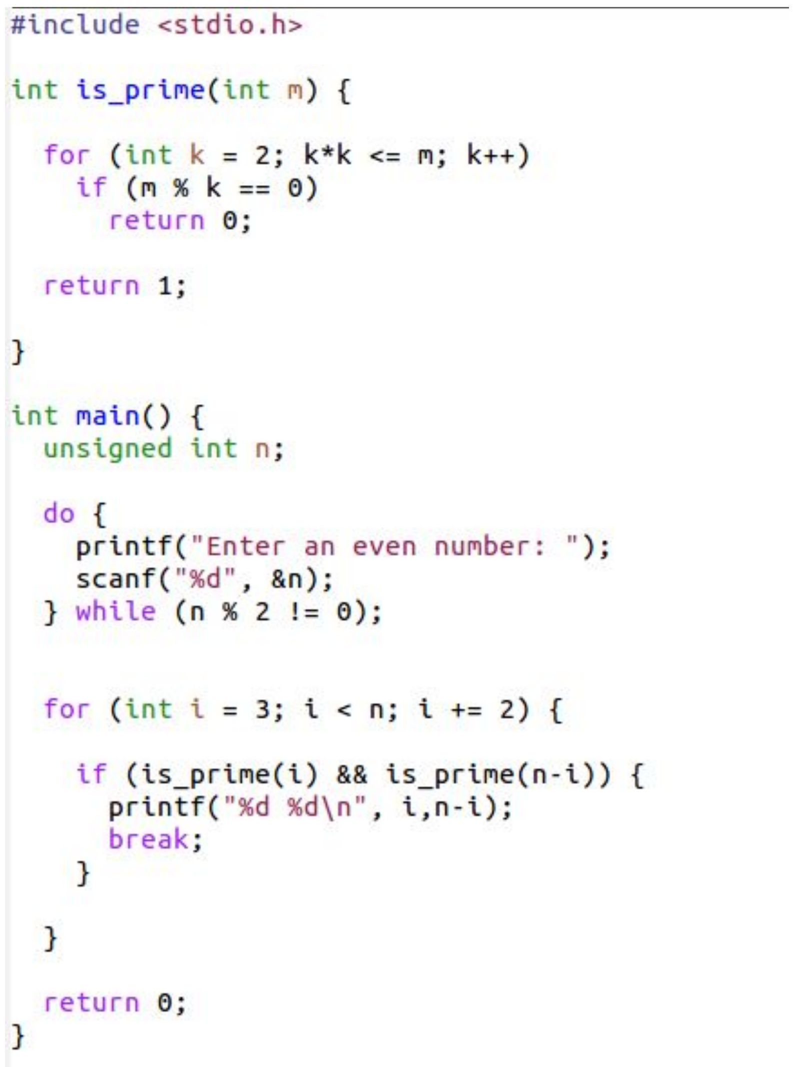
A function for detecting prime numbers



```
for (int i = 3; i < n; i += 2) {

  if (is_prime(i) && is_prime(n-i)) {
    printf("%d %d\n", i,n-i);
    break;
  }

}
```

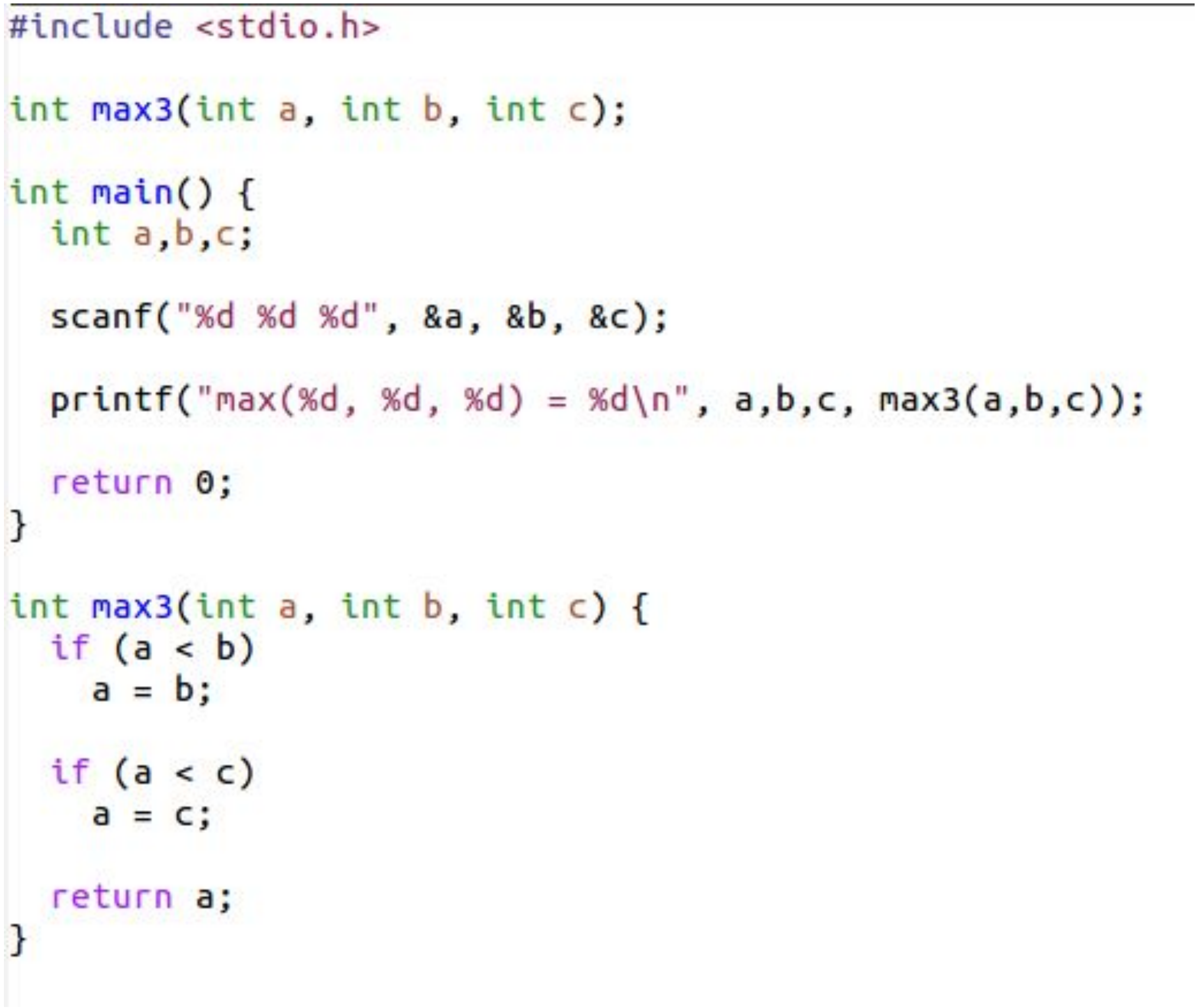


```
int is_prime(int m) {

  for (int k = 2; k*k <= m; k++)
    if (m % k == 0)
      return 0;

  return 1;

}
```

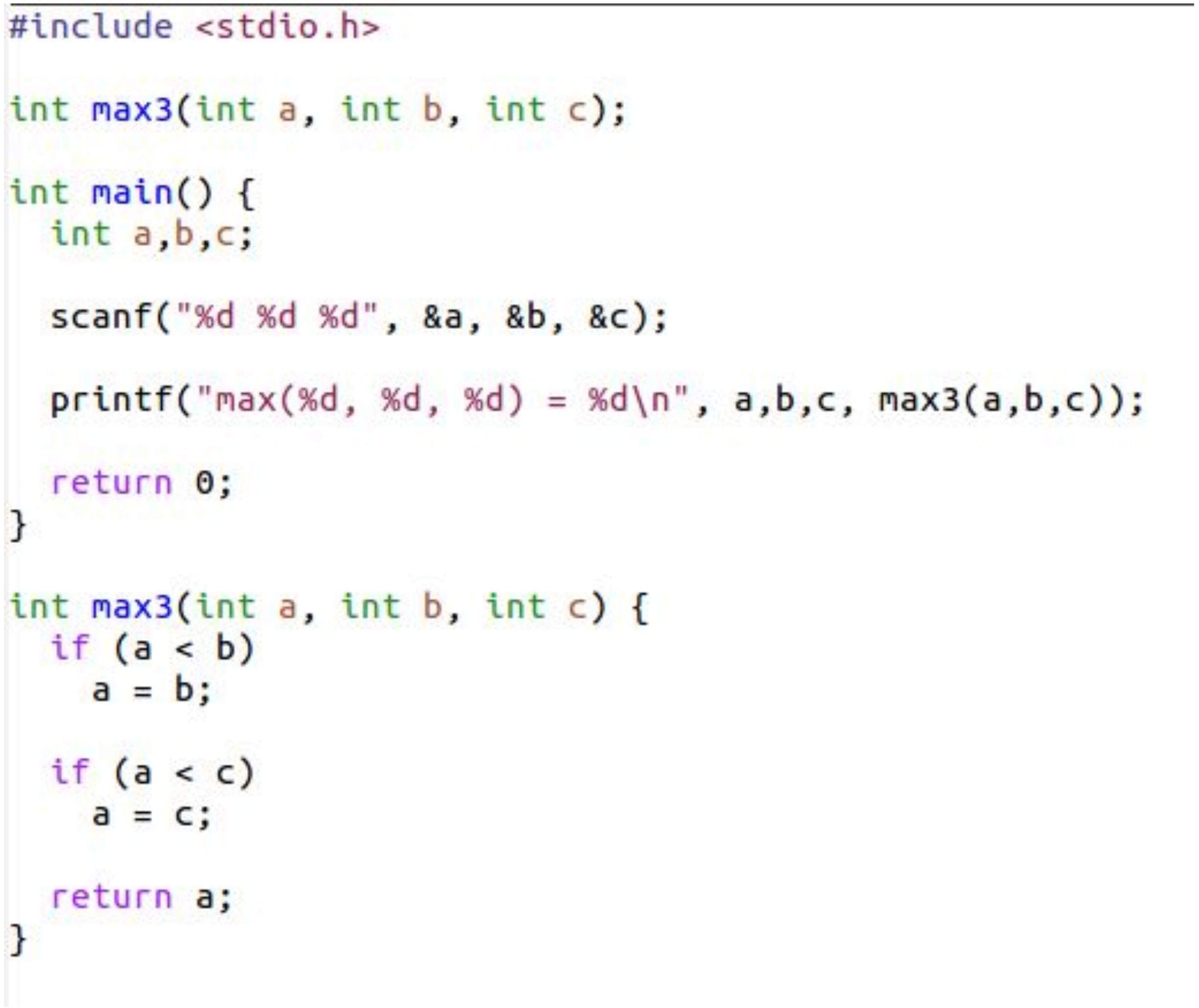


```c
#include <stdio.h>

int is_prime(int m) {

  for (int k = 2; k*k <= m; k++)
    if (m % k == 0)
      return 0;

  return 1;

}

int main() {
  unsigned int n;

  do {
    printf("Enter an even number: ");
    scanf("%d", &n);
  } while (n % 2 != 0);


  for (int i = 3; i < n; i += 2) {

    if (is_prime(i) && is_prime(n-i)) {
      printf("%d %d\n", i,n-i);
      break;
    }

  }

  return 0;
}
```

```c
#include <stdio.h>

int max3(int a, int b, int c);

int main() {
  int a,b,c, mx;

  scanf("%d %d %d", &a, &b, &c);

  mx = max3(a,b,c);

  printf("max(%d, %d, %d) = %d\n", a,b,c,mx);

  return 0;
}

int max3(int a, int b, int c) {
  if (a < b)
    a = b;

  if (a < c)
    a = c;

  return a;
}
```

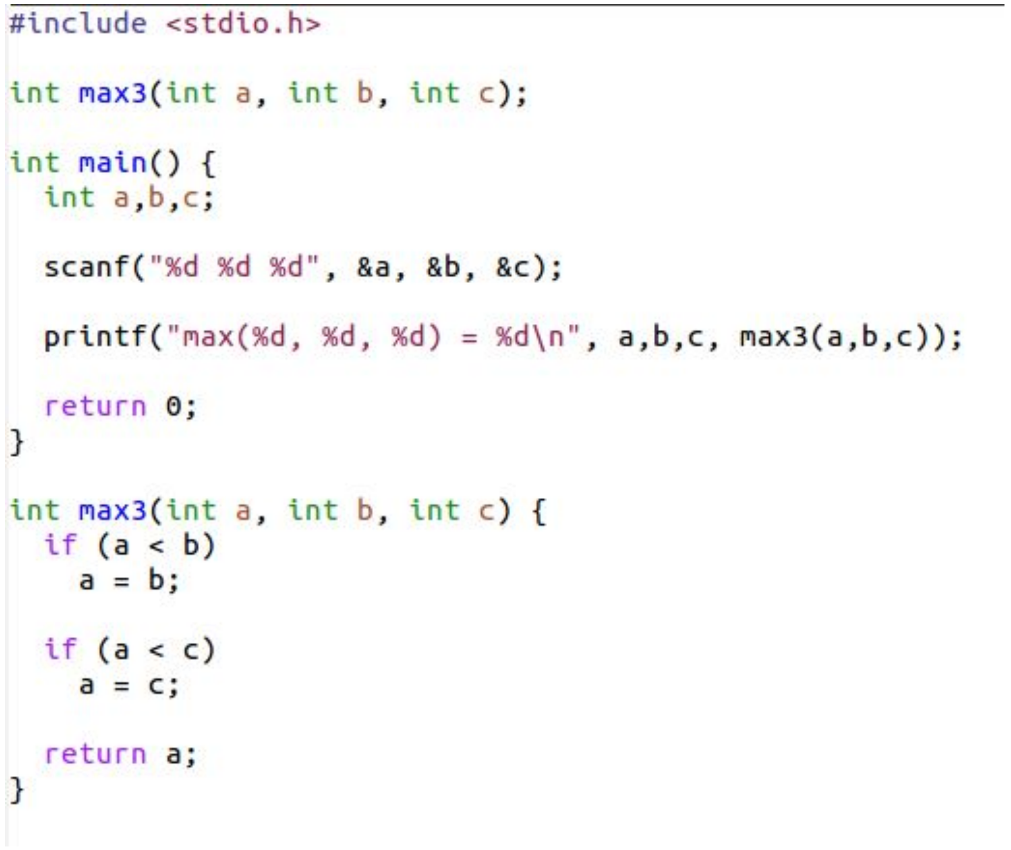


```c
#include <stdio.h>

int max3(int a, int b, int c);

int main() {
  int a,b,c;

  scanf("%d %d %d", &a, &b, &c);

  printf("max(%d, %d, %d) = %d\n", a,b,c, max3(a,b,c));

  return 0;
}

int max3(int a, int b, int c) {
  if (a < b)
    a = b;

  if (a < c)
    a = c;

  return a;
}
```



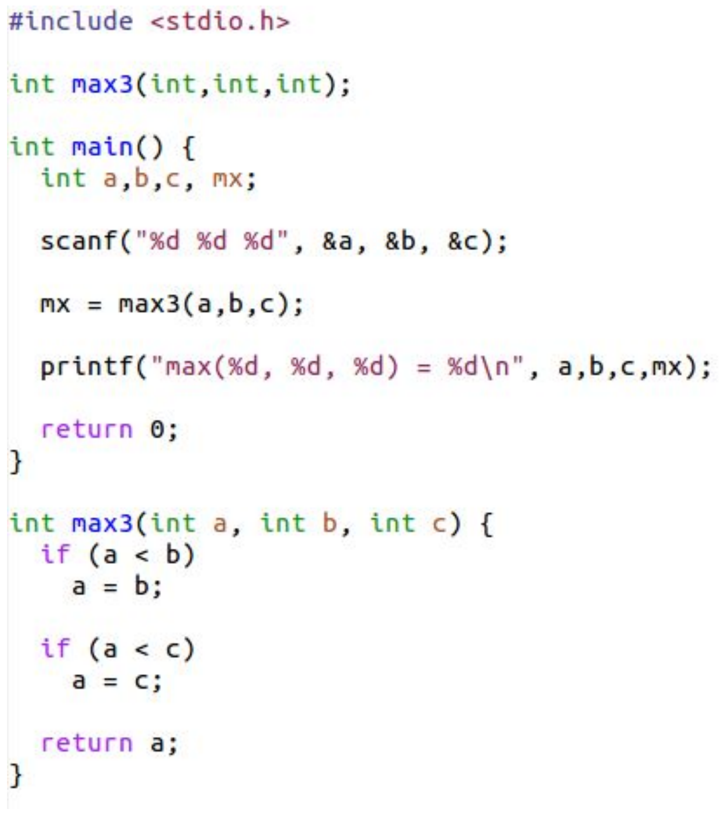
```c
#include <stdio.h>

int max3(int a, int b, int c);

int main() {
  int a,b,c;

  scanf("%d %d %d", &a, &b, &c);

  printf("max(%d, %d, %d) = %d\n", a,b,c, max3(a,b,c));

  return 0;
}

int max3(int a, int b, int c) {
  if (a < b)
    a = b;

  if (a < c)
    a = c;

  return a;
}
```

declarations / function prototypes

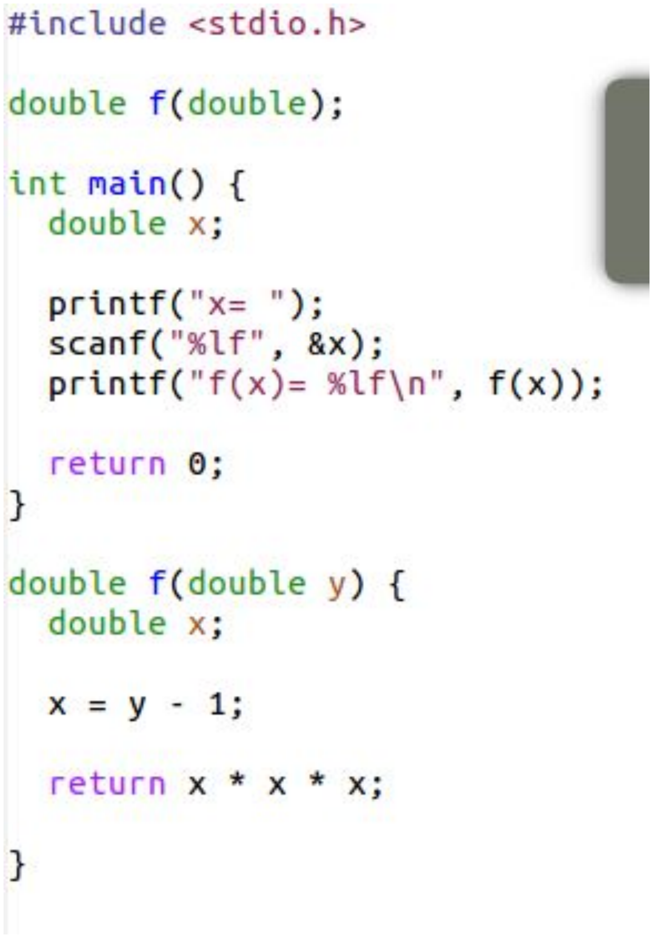


```
#include <stdio.h>

int max3(int a, int b, int c);

int main() {
  int a,b,c;

  scanf("%d %d %d", &a, &b, &c);

  printf("max(%d, %d, %d) = %d\n", a,b,c, max3(a,b,c));

  return 0;
}

int max3(int a, int b, int c) {
  if (a < b)
    a = b;

  if (a < c)
    a = c;

  return a;
}
```

declarations / function prototypes



```c
#include <stdio.h>

int max3(int,int,int);

int main() {
  int a,b,c, mx;

  scanf("%d %d %d", &a, &b, &c);

  mx = max3(a,b,c);

  printf("max(%d, %d, %d) = %d\n", a,b,c,mx);

  return 0;
}

int max3(int a, int b, int c) {
  if (a < b)
    a = b;

  if (a < c)
    a = c;

  return a;
}
```



```c
#include <stdio.h>

double f(double);

int main() {
  double x;

  printf("x= ");
  scanf("%lf", &x);
  printf("f(x)= %lf\n", f(x));

  return 0;
}

double f(double y) {
  double x;

  x = y - 1;

  return x * x * x;

}
```

C Preprocessor, include files

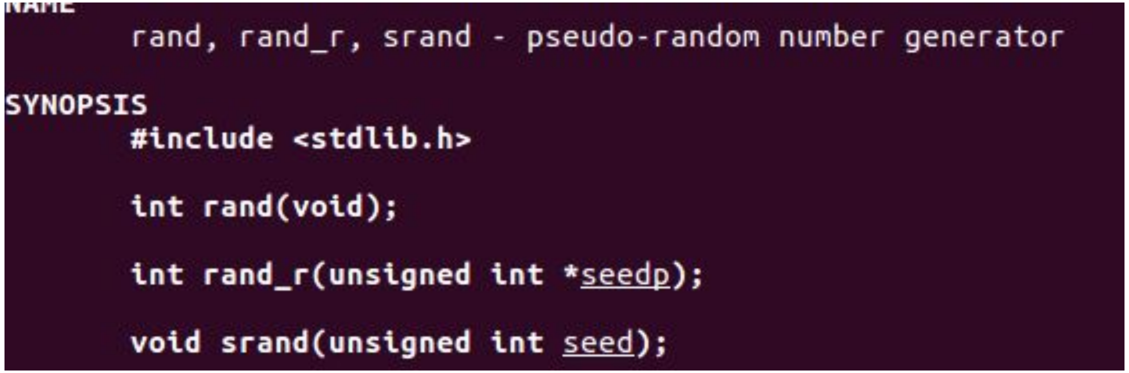
- printf (scanf, …)
  - library: **/lib/x86_64-linux-gnu/libc.so.6**
  - header file: **/usr/include/stdio.h**
    - **mostly contains function declarations**

C Preprocessor

- look at the header file
  - cat /usr/include/stdio.h
- look at the output of preprocessor:
  - gcc -E test.c

random number generation, rand, srand

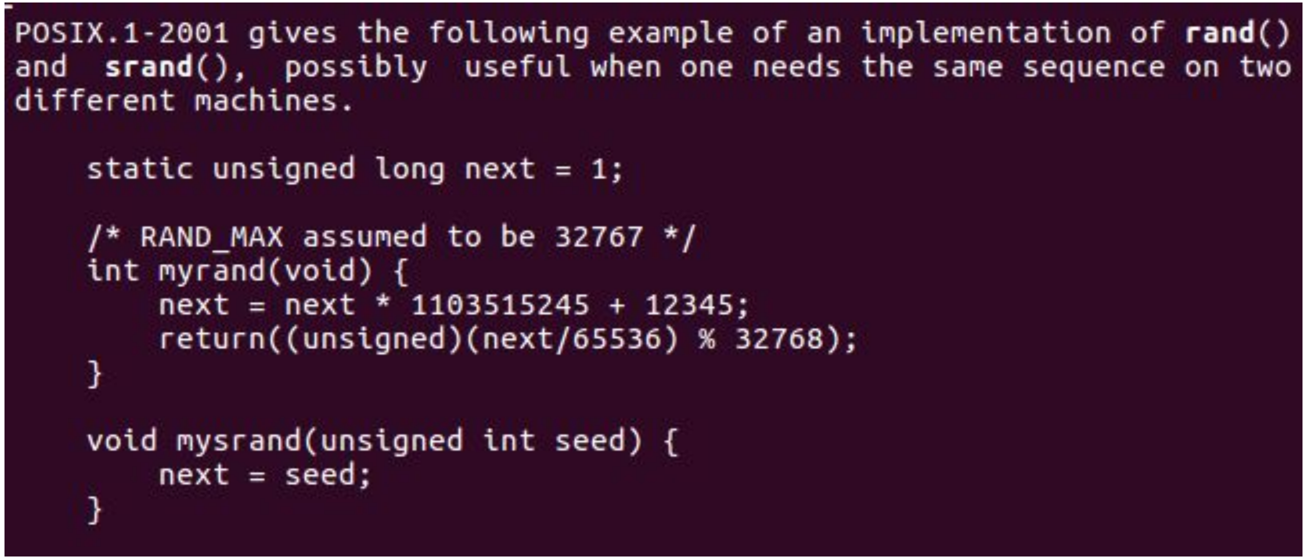
- run
  - man 3 rand



```
NAME
       rand, rand_r, srand - pseudo-random number generator

SYNOPSIS
       #include <stdlib.h>

       int rand(void);

       int rand_r(unsigned int *seedp);

       void srand(unsigned int seed);
```

random number generation, rand, srand

- also in **man 3 rand**



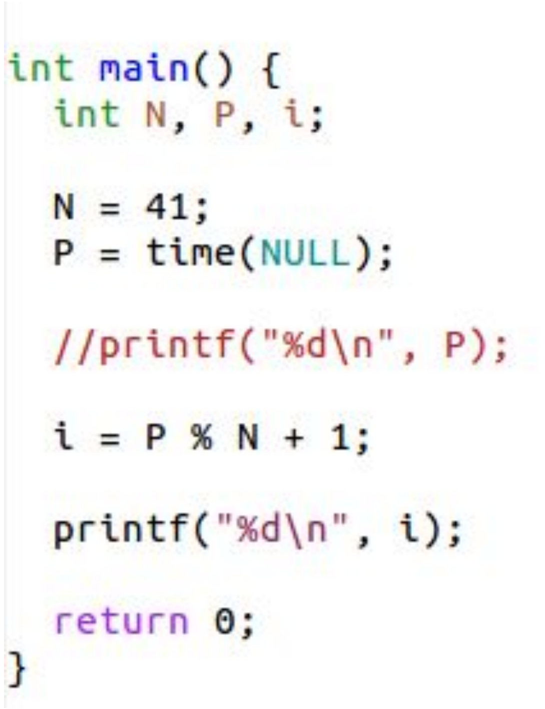
```
POSIX.1-2001 gives the following example of an implementation of rand()
and  srand(),  possibly  useful when one needs the same sequence on two
different machines.

    static unsigned long next = 1;

    /* RAND_MAX assumed to be 32767 */
    int myrand(void) {
        next = next * 1103515245 + 12345;
        return((unsigned)(next/65536) % 32768);
    }

    void mysrand(unsigned int seed) {
        next = seed;
    }
```

random number generation, rand, srand

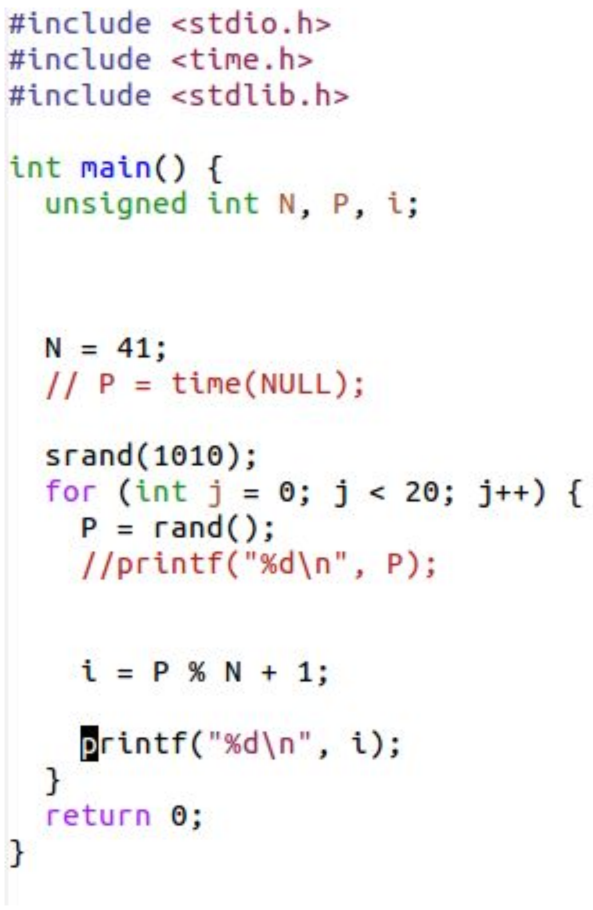
- a random number between **0** and **`RAND_MAX`**
  - `rand()`
- a random number between **0** and **n-1**
  - `rand() % n`
- a random number between **1** and **n**
  - `rand() % n + 1`

- **`RAND_MAX is defined in stdlib.h`**
  - `cat /usr/include/stdlib.h`
  - `#define  RAND_MAX    2147483647`

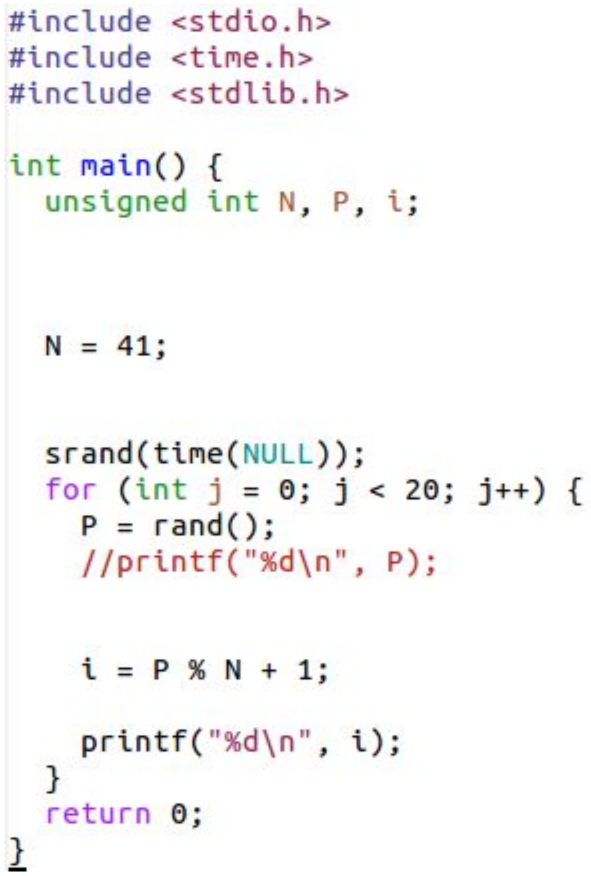
remember choose.c



```
int main() {
  int N, P, i;

  N = 41;
  P = time(NULL);

  //printf("%d\n", P);

  i = P % N + 1;

  printf("%d\n", i);

  return 0;
}
```



```c
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int main() {
  unsigned int N, P, i;



  N = 41;
  // P = time(NULL);

  srand(1010);
  for (int j = 0; j < 20; j++) {
    P = rand();
    //printf("%d\n", P);


    i = P % N + 1;

    printf("%d\n", i);
  }
  return 0;
}
```



```c
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int main() {
  unsigned int N, P, i;



  N = 41;


  srand(time(NULL));
  for (int j = 0; j < 20; j++) {
    P = rand();
    //printf("%d\n", P);


    i = P % N + 1;

    printf("%d\n", i);
  }
  return 0;
}
```