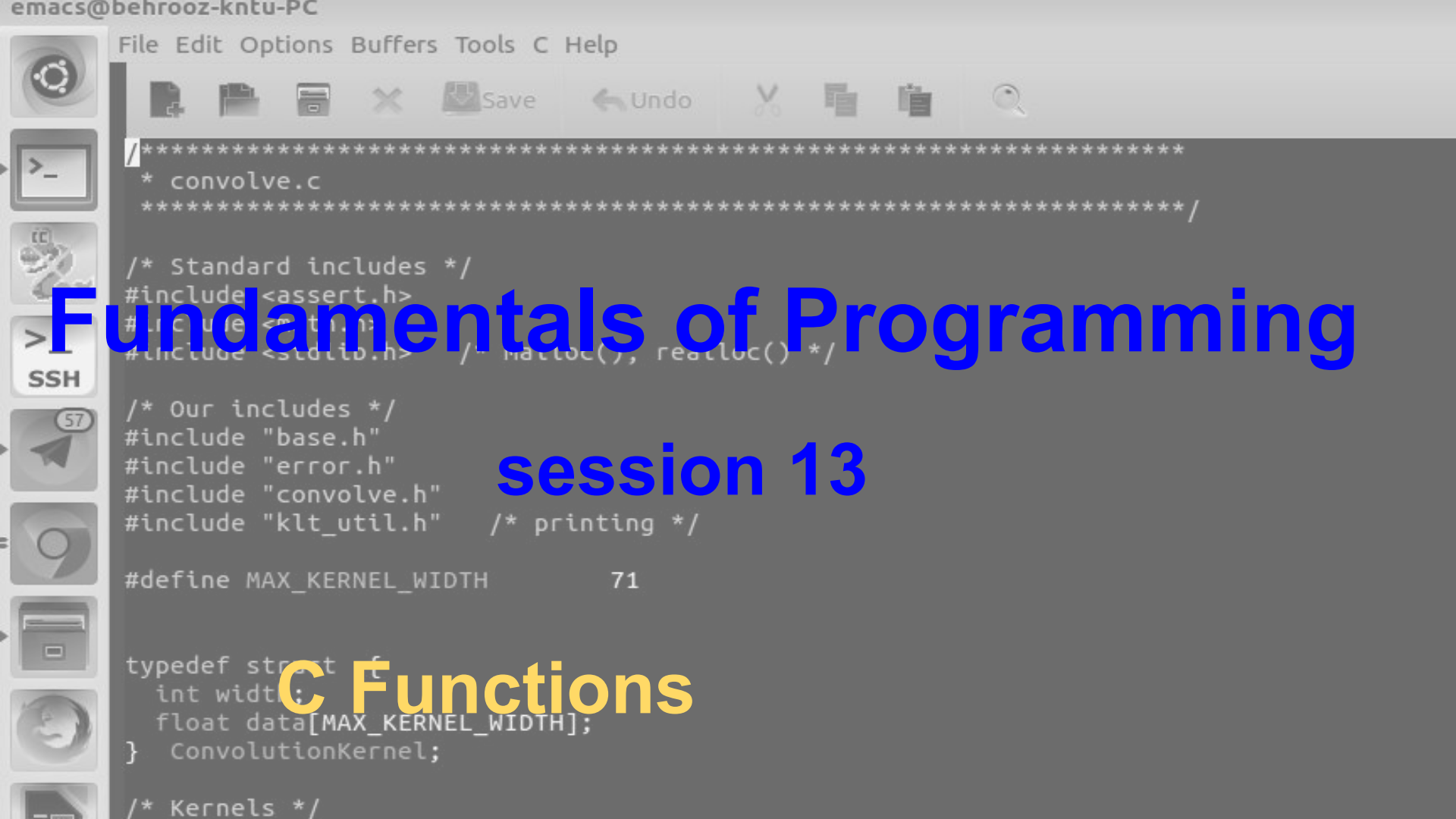

emacs@behrooz-kntu-PC
File Edit Options Buffers Tools C Help
Save
Undo
/* convolve.c */
/* Standard includes */
#include <assert.h>
/* Our includes */
#include "base.h"
#include "error.h"
#include "convolve.h"
#include "klt_util.h" /* printing */
#define MAX_KERNEL_WIDTH 71
typedef struct {
int width;
float data[MAX_KERNEL_WIDTH];
} ConvolutionKernel;
/* Kernels */
Fundamentals of Programming
session 13
C Functions

```c
int g,a;

a = 0;
while (1) {
  printf("Enter grade: ");
  scanf("%d", &g);

  if (g== -1)
    break;

  a = a || g < 10;

}


if (a)
  puts("fail");
else
  puts("pass");
```

```c
int g,a;

a = 0;
while (1) {
  printf("Enter grade: ");
  scanf("%d", &g);

  if (g== -1)
    break;

  a = a || (g < 10);

}


if (a)
  puts("fail");
else
  puts("pass");
```

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**
  - **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

```c
#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

```c
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

```
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

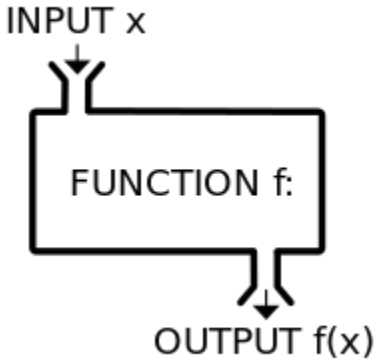
Comments!

```
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;
  }

}
```

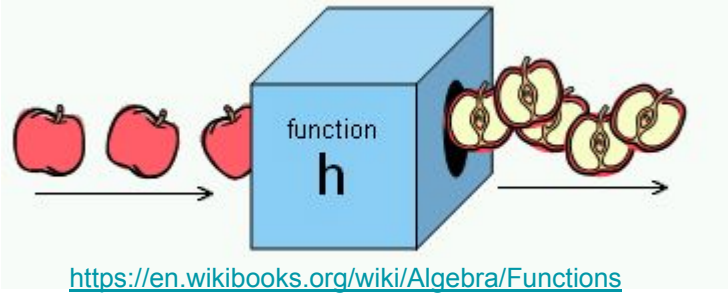
What's wrong?

```
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;
  }

}
```

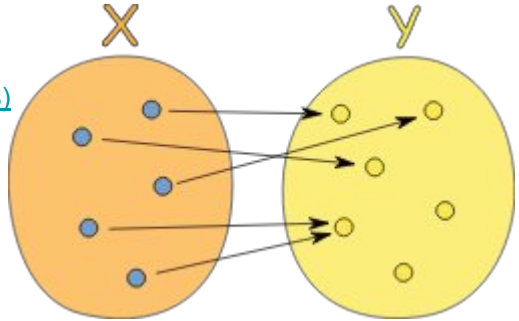
Functions

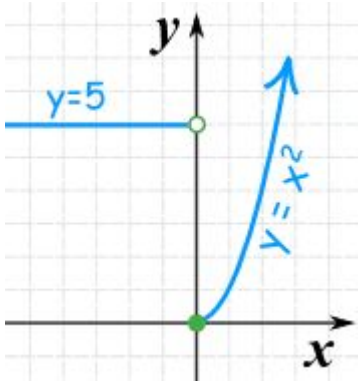


https://en.wikipedia.org/wiki/Function_(mathematics)



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





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

```
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) {
  int j = n - i;

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

}
```

```
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) {
  int j = n - i;

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

}
```

```
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;
}
```