CPS 188

Computer Programming Fundamentals

Prof. Alex Ufkes

Topic 6.2: Pointers & examples

Toronto Metropolitan University

Notice!

Obligatory copyright notice in the age of digital delivery and online classrooms:

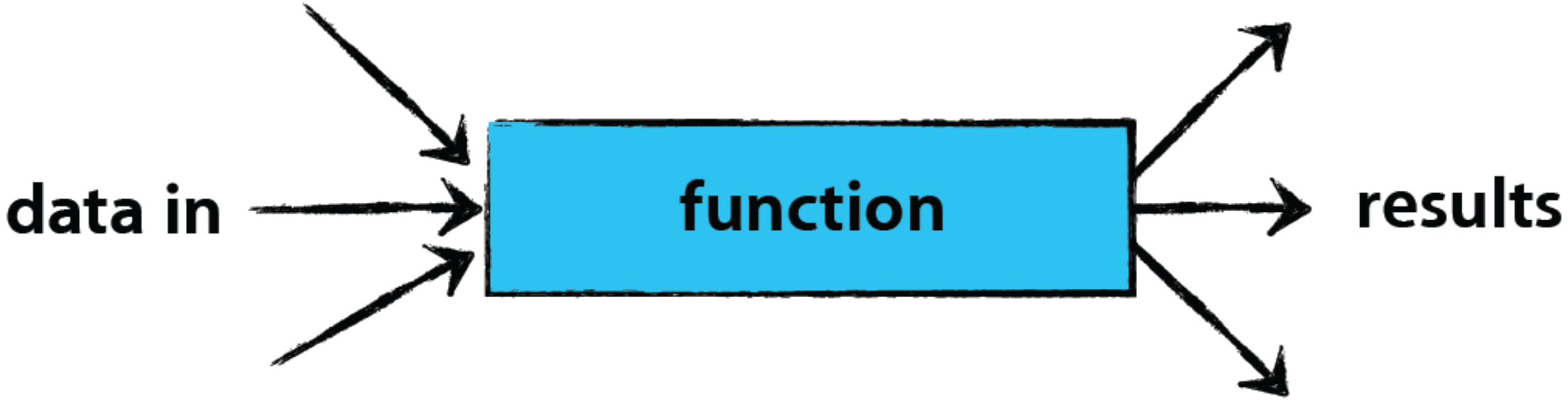
Today

Assorted Examples

POINTERS

Multiple results?

(WITHOUT using global variables!)



POINTERS to the rescue!

quad()

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

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

double quad (double a, double b, double c)
{
   double disc = sqrt(b*b – 4*a*c);
   double root1 = (-b + disc)/(2*a);
   double root2 = (-b - disc)/(2*a);
   return root1; // What about root2?
}

int main (void)
{
   double c1=1, c2=2, c3=3;
   double r1 = quad(c1, c2, c3);
   printf("root1 = %.2lf", r1);
   return 0;
}
```

"Can we have multiple results?"

NO.

We're stuck!

- One option? Two separate functions, one for each root.
- Another option? Pointers, which we haven't learned yet.

Quadratic Formula

```
void quadForm (double a, double b, double c,
                                double *x1, double *x2)
{
      double tmp = sqrt(b*b – 4*a*c);

      *x1 = (-b + tmp)/(2*a);
      *x2 = (-b - tmp)/(2*a);
}
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Quadratic Formula

```
void quadForm (double a, double b, double c,
               double *x1, double *x2);

int main (void)
{
    double a = 1.1, b = 7.8, c = 2.5;
    double root1, root2;

    quadForm(a, b, c, &root1, &root2);
    printf("roots: %lf, %lf", root1, root2);

    return 0;
}
```

quad_form.c - C:\Users\aufke\Google Drive\Teaching\CPS 188\Code Samples - Geany

File Edit Search View Document Project Build Tools Help

Symbols

- Functions
  - main [11]
  - quadForm [4]

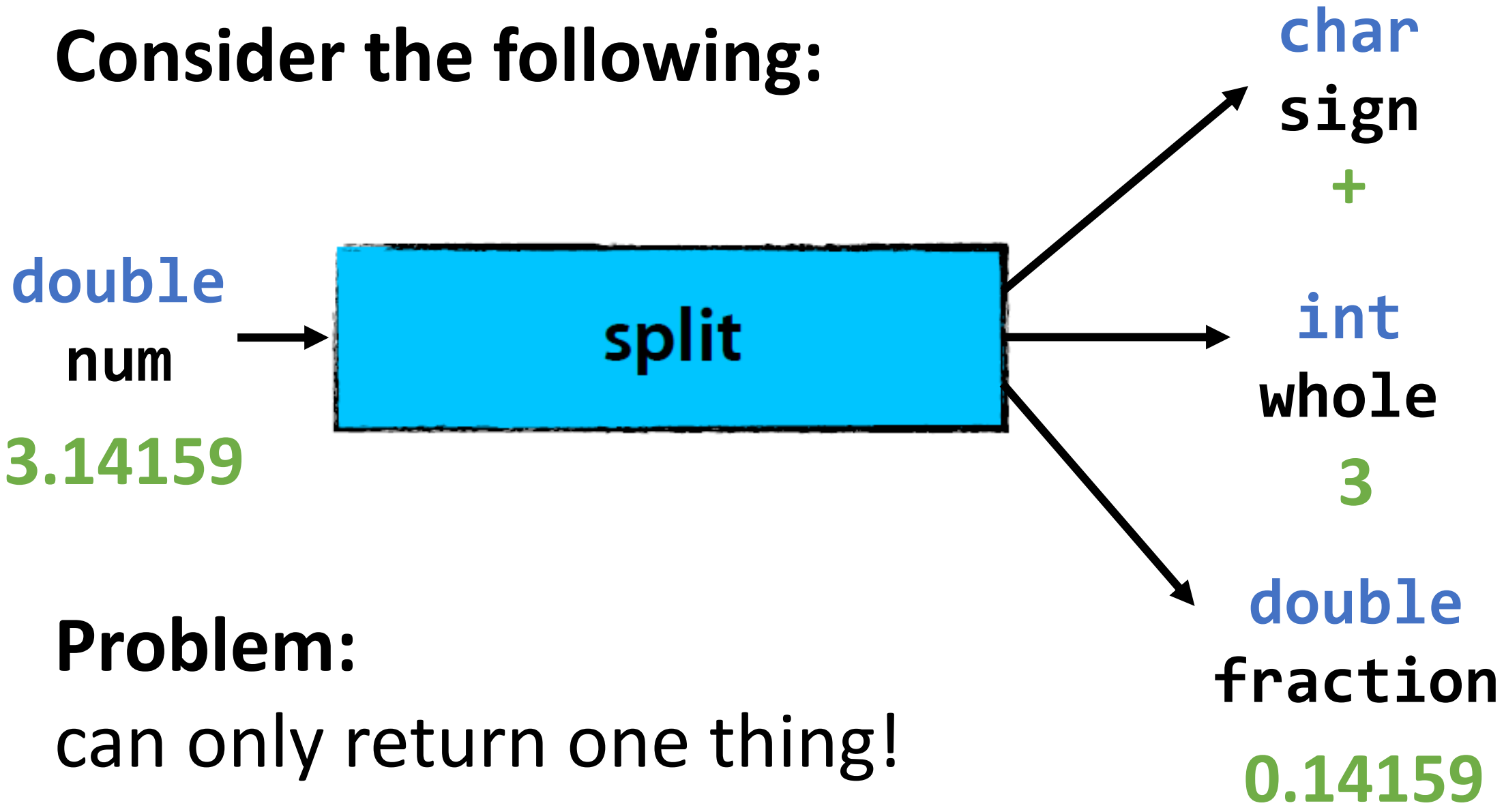
quad_form.c

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

void quadForm (double a, double b, double c, double *x1, double *x2)
{
    double tmp = sqrt(b*b - 4*a*c);
    *x1 = (-b + tmp)/(2*a);
    *x2 = (-b - tmp)/(2*a);
}

int main (void)
{
    double a = 1.1, b = 7.8, c = 2
    double root1, root2;

    quadForm(a, b, c, &root1, &roo
    printf("roots: %lf, %lf", root

    return 0;
}
```

C:\WINDOWS\SYSTEM32\cmd.exe

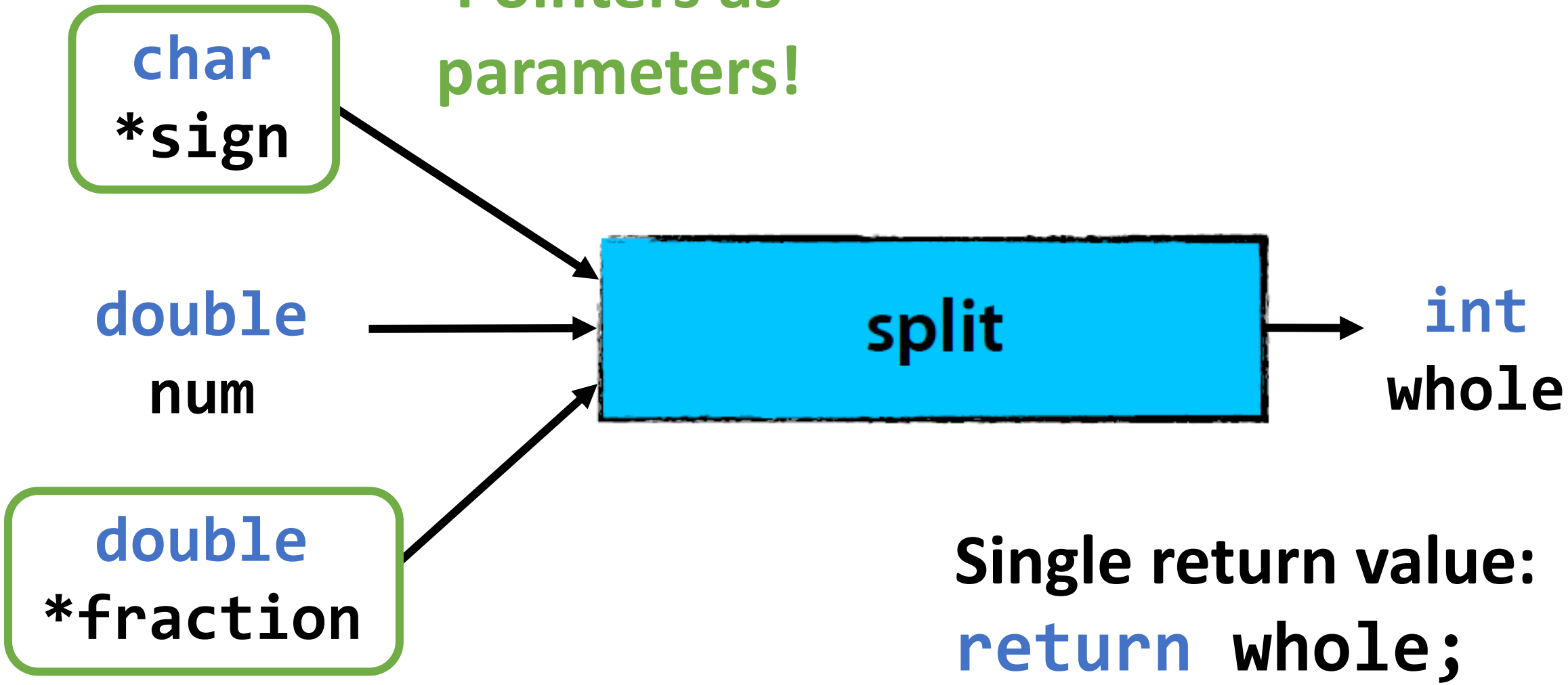
```
roots: -0.336480, -6.754430

------------------
(program exited with code: 0)

Press any key to continue . . .
```

Consider the following:



Problem:
can only return one thing!

Pointers as parameters!



Single return value:

```
return whole;
```

```
int split (double num, char *sign, double *fraction)
{
    int whole = abs((int)num);
    *fraction = fabs(num) - whole;
    if (num >= 0)
        *sign = '+';
    else
        *sign = '-';
    return (whole);
}
```

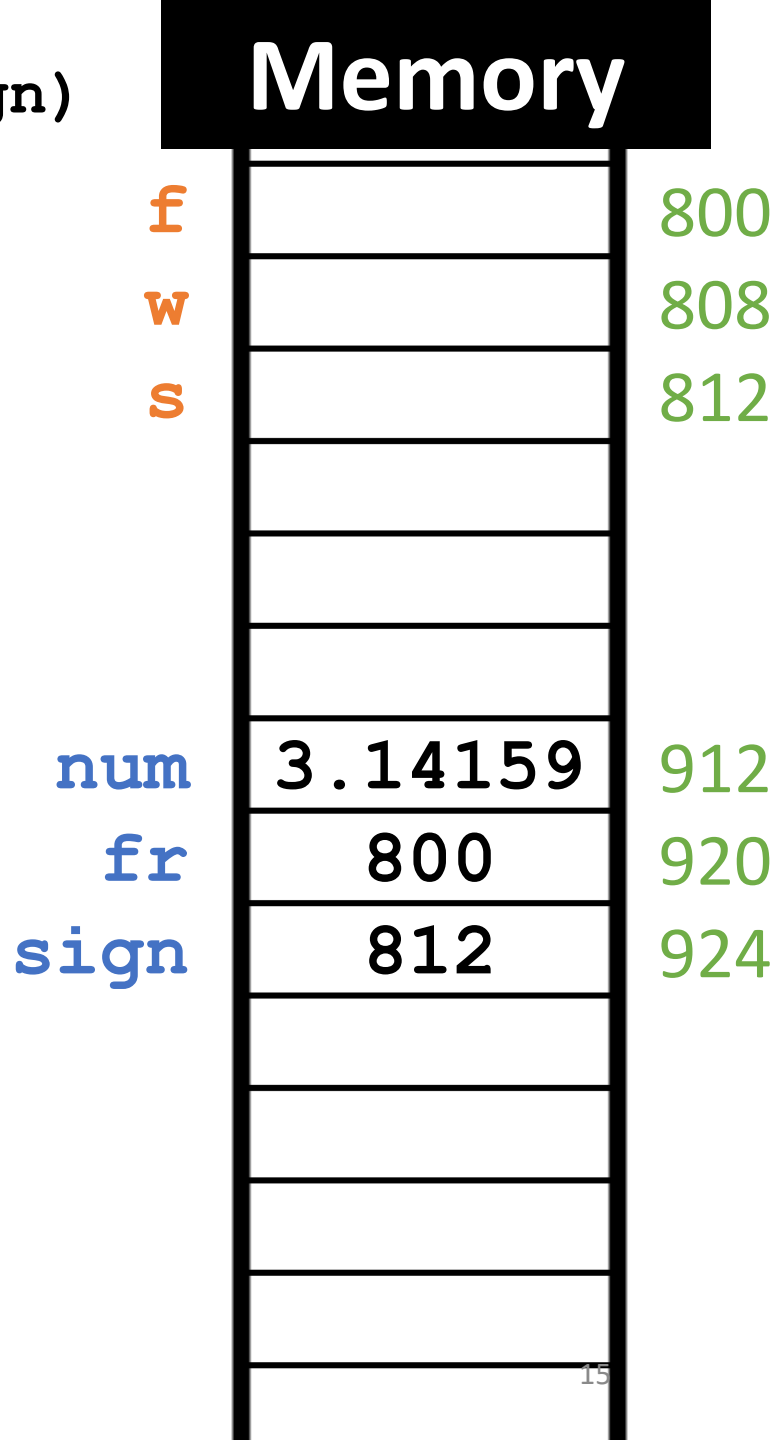
Pointers

De-reference pointers to assign values

```c
int split (double num, char *sign, double *fraction)
{ /*split code*/
}

int main (void)
{
    double f, n = 3.1415;
    char s;

    int w = split(n, &s, &f);

    printf("Sign:     %c\n", s);
    printf("Whole:    %d\n", w);
    printf("Fraction: %lf\n", f);

    return (0);
}
```

When we de-reference `sign` and `fraction` in `split`, we are accessing `s` and `f` in `main`!

```
int split (double num, double *fr, char *sign)
{
    int whole = abs((int)num);
    *fr = fabs(num) - whole;
    if (num >= 0)   *sign = '+';
    else            *sign = '-';
    return (whole);
}

int main (void)
{
    double f; int w; char s;
    w = split(3.14159, &f, &s);
    printf("%c %d %lf", s, w, f);
    return (0);
}
```



```
int split (double num, double *fr, char *sign)
{
    int whole = abs((int)num);  /* 3 */
    *fr = fabs(num) - whole;  /* 0.14159 */
    if (num >= 0)   *sign = '+';
    else   *sign = '-';
    return (whole);
}

int main (void)
{
    double f; int w; char s;
    w = split(3.14159, &f, &s);
    printf("%c %d %lf", s, w, f);
    return (0);
}
```

Memory

f	0.14159	800
w	3	808
s	'+'	812
num	3.14159	912
fr	800	920
sign	812	924
whole	3	928

Swap Two Variables

Very common operation!

```
void swap (int x, int y)
{
    x = y;
    y = x;
}
```

Is there a problem here?

Swap Two Variables

Very common operation!

```
void swap (int x, int y)
{
    int tmp = x;
    x = y;
    y = tmp;
}
```

How about now?

Swap Two Variables

```
void swap (int x, int y)
{
    int tmp = x;
    x = y;
    y = tmp;
}
```

```
int main (void)
{
    int a = 1, b = 2;
    printf("%d, %d\n", a, b);
    swap(a, b);
    printf("%d, %d\n", a, b);
    return 0;
}
```

What prints?

swap.c - C:\Users\aufke\Google Drive\Teaching\CPS 188\Code Samples - Geany

File Edit Search View Document Project Build Tools Help

Symbols

swap.c

- Functions
  - main [10]
  - swap [3]

```c
#include <stdio.h>

void swap (int x, int y)
{
    int tmp = x;
    x = y;
    y = tmp;
}

int main (void)
{
    int a = 1, b = 2;
    printf("%d, %d\n", a, b);
    swap(a, b);
    printf("%d, %d\n", a, b);
    return 0;
}
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
1, 2
1, 2


------------------
(program exited with code: 0)

Press any key to continue . . .
```

Swap Two Variables

```c
void swap (int *x, int *y)
{
    int tmp = *x;
    *x = *y;
    *y = tmp;
}
```

```c
int main (void)
{
    int a = 1, b = 2;
    printf("%d, %d\n", a, b);
    swap(&a, &b);
    printf("%d, %d\n", a, b);
    return 0;
}
```

What prints?

swap.c - C:\Users\aufke\Google Drive\Teaching\CPS 188\Code Samples - Geany

File Edit Search View Document Project Build Tools Help

Symbols

- Functions
  - main [10]
  - swap [3]

swap.c

```c
#include <stdio.h>

void swap (int *x, int *y)
{
    int tmp = *x;
    *x = *y;
    *y = tmp;
}

int main (void)
{
    int a = 1, b = 2;
    printf("%d, %d\n", a, b);
    swap(&a, &b);
    printf("%d, %d\n", a, b);
    return 0;
}
```

C:\WINDOWS\SYSTEM32\cmd.exe

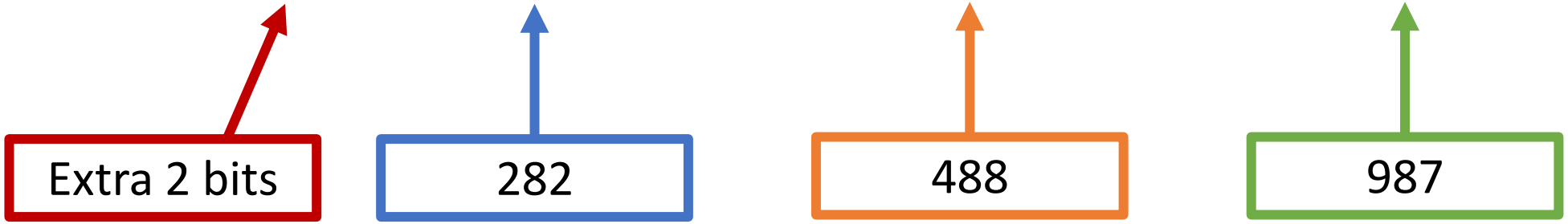
```
1, 2
2, 1


------------------
(program exited with code: 0)

Press any key to continue . . .
```

Recall: Integer Packing

Shift 10 bits at a time:

- Consider three numbers: **282**, **488**, **987**
- We want to pack them into a single **`int`** like so:

00010001101001111010001111011011



intpack.c - C:\Users\aufke\Google Drive\Teaching\CPS 188\Code Samples - Geany

File Edit Search View Document Project Build Tools Help

Symbols

- Functions
  - main [17]
  - pack3 [3]
  - unpack3 [10]

intpack.c

```c
#include <stdio.h>

int pack3 (int a, int b, int c)
{
    int packed = a << 10;
    packed = (packed | b) << 10;
    return packed | c;
}

void unpack3 (int packed)
{
    printf("%d\n", packed & 0x3FF);
    printf("%d\n", (packed >> 10) & 0x3FF);
    printf("%d\n", (packed >> 20) & 0x3FF);
}

int main (void)
{
    int packed = pack3(282, 488, 987);
    printf("packed = %d\n", packed);
    unpack3(packed);
    return 0;
}
```

- Instead of printing in the function (usually bad)…
- Let's use pointers to send back all three values.

```c
void unpack3 (int packed)
{
    printf("%d\n", packed & 0x3FF);
    printf("%d\n", (packed >> 10) & 0x3FF);
    printf("%d\n", (packed >> 20) & 0x3FF);
}
```

```c
void unpack3 (int packed, int *a, int *b, int* c)
{
    *a = packed & 0x3FF;
    *b = (packed >> 10) & 0x3FF;
    *c = (packed >> 20) & 0x3FF;
}
```

swap.c intpack.c

```c
#include <stdio.h>

int pack3 (int a, int b, int c)
{
    int packed = a << 10;
    packed = (packed | b) << 10;
    return packed | c;
}

void unpack3 (int packed, int *a, int *b, int* c)
{
    *a = packed & 0x3FF;
    *b = (packed >> 10) & 0x3FF;
    *c = (packed >> 20) & 0x3FF;
}

int main (void)
{
    int n1, n2, n3;
    int packed = pack3(282, 488, 987);
    printf("packed   = %d\n", packed);
    unpack3(packed, &n1, &n2, &n3);
    printf("unpacked = %d, %d, %d\n", n1, n2, n3);
    return 0;
}
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
packed   = 296199131
unpacked = 987, 488, 282


------------------
(program exited with code: 0)

Press any key to continue . . .
```

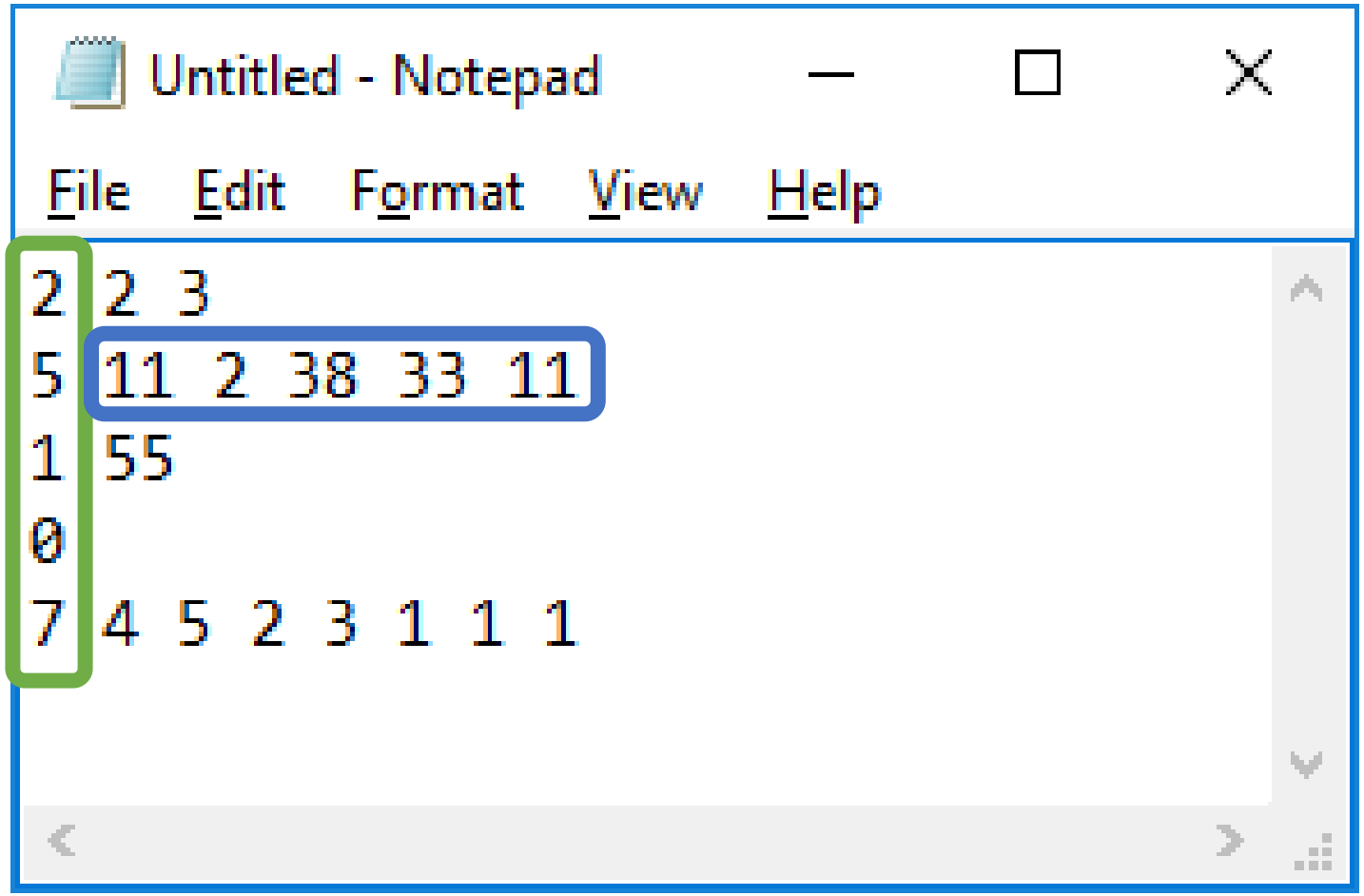
Nested Loops

+

File I/O

Write a code snippet to do the following:

Compute the averages of the integers in each row in a file. The first number in each row indicates the number of integers on that row.



Write a code snippet to do the following:

Compute the averages of the integers in each row in a file. The first number in each row indicates the number of integers on that row.

```
int i, value, sum;
int num_in_row;
double avg;
FILE *in = fopen("data.txt", "r");

while (fscanf(in, "%d", &num_in_row) != EOF)
{
        sum = 0;
        for (i = 0; i < num_in_row; i++) {
                fscanf(in, "%d", &value);
                sum += value;
        }
        avg = (double)sum/num_in_row;
        printf("Row average = %6.2lf \n", avg);
}
```

Read number of integers per row

Initialize sum for current row

Read all integers on current row

Compute and print average

Repeat until EOF

Did your file open correctly?

`fopen` isn't guaranteed to work

If your program involving file I/O ***compiles*** but doesn't work as expected, your file may **not** have opened successfully:

```
FILE *in = fopen("data.txt", "r");
if (in == NULL) /* File didn't open correctly */
    printf("File failed to open!\n");
```

Double check file name, location, etc.

Spot the ERROR?

```
#include <stdio.h>

int main (void)
{
        int x, y = 2;
        int* w, *z;

        *w = 1;
        *x = 125;

        p

        return 3-2;
}
```

Pointer not initialized

Primitives cannot be dereferenced

```
int main (void)
{
    int x, y, z;
    int *p1, *p2, p3*; char * p4;

    p1 = &x;
    x = 125;
    p2 = p1;
    p4 = p2;


    return 125;
}
```

Incompatible pointer types

```c
#include <stdio.h>

int main(void)
{
    int n, x = -3;
    for (n = x, n <= 0, n += 1)
        printf("%d\n", n);

    return 0;
}
```

Should be semicolons, not commas

```c
#include <stdio.h>

int main(void)
{
    int n;
    for (n = 1; n != 50; n *= 2) {
        printf("%d\n", n);
    }

    return 0;
}
```

```
#include <stdio.h>

int main (void)
{
    int x = 5;
    while( x > 0 );
        x--;

    return 55;
}
```

```
#include <stdio.h>

int main(void)
{
    for (i = 1; i <= 12; i++)
    {
        printf("%d\n",i);
    }

    return (0);
}
```

Variable not declared!

```c
#include <stdio.h>

int main(void)
{
    int n;

    do
    {
        printf("Enter a number ");
        printf("between 1 and 9: ");
        scanf ("%d", &n);
    }while (n < 1 || n > 9)

  return (0);
}
```

Questions?