Lec Notes – W3

- 4c [20m]: saving.c – a detailed development for saving.c with complicated nested loop, reasons for using #define, also repeated about program presentation
- 4d [26m]: the wile loop; then
  - threen.c (input n; while (n>1) { if (n odd) n= 3n+1; else n /=2; }
  - threenbig.c (adding counter of num of iterations)
  - isprime.c : loop with break; improve speed with d*d<=n
- 4e [25m]: using scanf to control loop, readloop1.c then
  - fortcomm.c (discard fortran comments), ridirection ./fortcomm < data.txt >output.txt. Commands more, ls, rm

Chapter 5: Functions ➔ We'll do this one (and lec W4) in the next workshop

- 5a [26m]: intro, isprimefunc.c where using function int isprime(int) then:
  - a bit about argc, argv; local variables
- 5b [20m]: abstraction with functions, mechanism, details on passing arg & return values; then
  - functions int_max_3, concept of scalfold
- 5c [17m]: compilation with functions
  - option 1: single .c file, structure of prog (with #define, functions…)
  - option 2 and 3: modules, not used in this course
  - library functions, stdio and stdlib; math.h & usemath.c; islower() and ctype.h

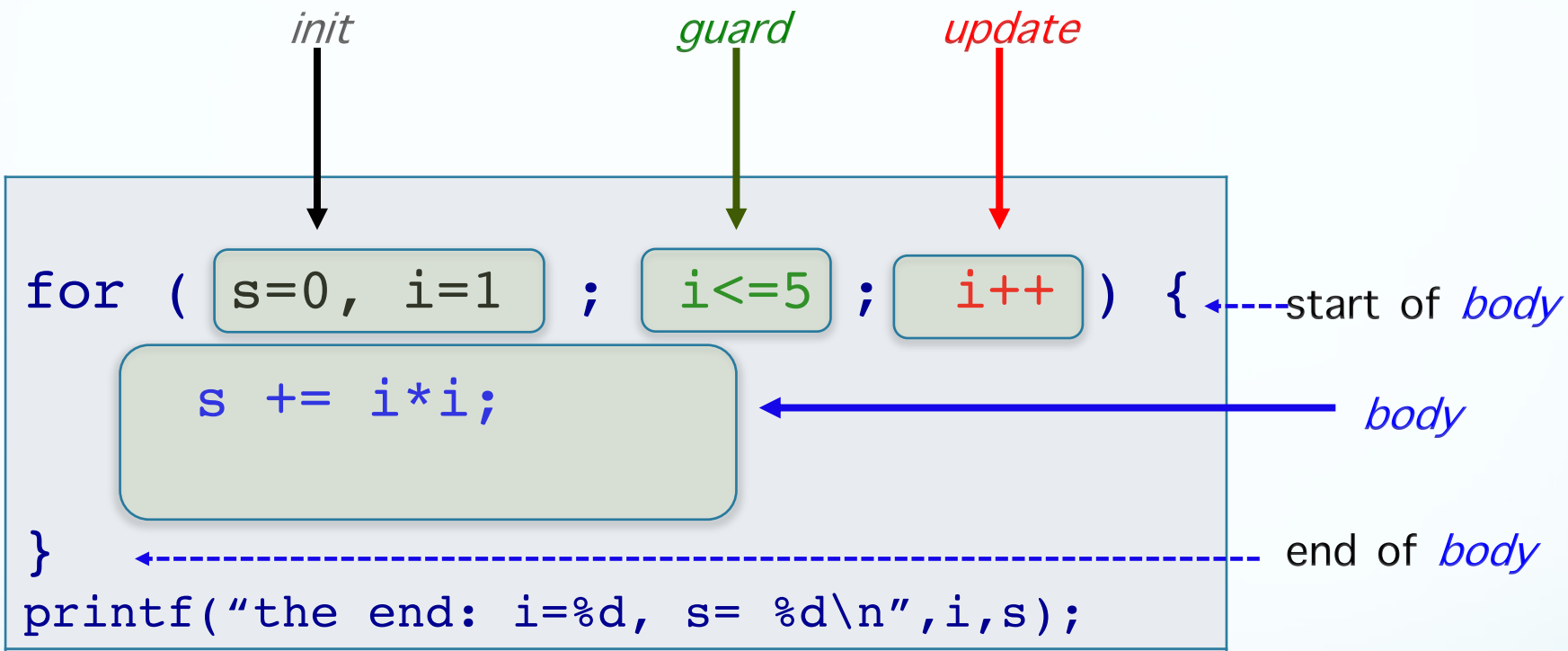
Today's Program

```
   int main(int argc, char *argv[]) {
1    discuss("loops: for, while, and do…while");
2    discuss("4.01 & 4.02");
3    design_and_implement("4.05");
4    have_fun("quiz");
5    implement("4.09");
6    implement("4.04");
7    for (i=0; i<4 && having_time(); i++) {
8      implement("next undone exercise from C04");
9    }
     return 0;
   }
```

Quiz 1: Wednesday 30 March, 4:15pm Melbourne time, covers chapters 1-5

Remember to try sample quiz *several times* before next workshop and bring questions to the class.

Loops



- Note: The comma operator can be used to combine a series of assignments into a single statement.

- The above loop is equivalent to

  s= 0 + 1*1 + 2*2 + 3*3 + 4*4 + 5*5

when i= 1 2 3 4 5 6

loops with no `init` and update

```
sum=0;
for (    ;  scanf("%d",&x)==1  ;    ) {
        sum += x;
}
printf("sum= %d\n", sum);
```

```
// equivalent to the top
sum=0;
while (   scanf("%d",&x)==1  ) {
        sum += x;
}
printf("sum= %d\n", sum);
```

continue & break: skip an iteration & early termination

```
for ( s=0, i=1 ; i<=10 ; i++ ) {
    if (i==2) {
        continue;
    }

    s += i*i;

    if (i==4) {
        break;
    }
}

printf("the end: i=%d, s= %d\n",i,s);
```

Loops: for, while and do…while

Compare:

```
while ( guard ) {
        statement;
}
```

and

```
do {
        statement;
} while (guard);
```

4.2

Give a general construction that shows how any **`do`** *statement can be converted into an equivalent* **`while`** *statement.*

```
do {
        statement;
} while (guard);
```

```
while ( guard ) {
        statement;
}
```

Examples: build code fragments for computing:

- $S= 1^2+ 2^2 + \ldots + n^2$
- $S= 1/1 + 1/2 + 1/3 + \ldots$

  the sum should include, and only include 1/i such that 1/i >= 0.0001
- $S= 1/1 + ½ + \ldots + 1/n + 2/1 + 2/2 + \ldots + 2/n + \ldots + n/1 + n/2 + \ldots + n/n$

4.1a

Trace the action of the loop, and determine the values printed out by the `printf` *statement. Assume that all variables have been declared to be of type* `int`.

```
10  ...
11  for (i=0; i<20;     i= i+3) {
12      printf ("%2d\n", i);
13  }
14  ...
```

output

4.1 extension

Trace the action of the loop, and determine the values printed out by the `printf` *statement. Assume that all variables have been declared to be of type* `int`.

```
1   for (i=1; i<5;    i++) {

2     for (j=1; j<=i; j++) {
3       printf ("i=%d, j=%d\n",i,j);
4     }
5     if (i==2) break;

6   }
    …
```

output

Now, in group, finish exercise 4.01 questions b-g

- Note:
  - When working in group, feel free to chat with your mates
  - Instructors will not enter your chat room
  - Your chat room will close after 10 minutes approximately, you will be notified one minute before the closing
  - We will go back to the whole-class mode after chat rooms closed

Work/Discuss with your friends
See questions one-by-one in `grok` C04.Exercise 4.05

```
-------- (b) ---------
for  (i=1; i<2000000; i= 2*i) {
     printf ("%7d\n", i);
}

-------- (c) ---------
sum = 0;
for  (i=0; i<10; i++) {
     sum = sum + i;
     printf ("S(%2d) = %2d\n", i,
sum);
}
```

Work/Discuss with your friends

```
-------- (d) --------
for (i= 0; i < 8; i++) {
  for  (j= i+1; j < 8; j += 3) {
    printf (“i= %d, j= %d\n”, i, j);
  }
}

-------- (e) ---------
for (i= 0; i < 8; i++) {
  for  (j= i+1; j < 8; j += 3) {
    if (i+j == 7) {
      break;
    }
    printf (“i= %d, j= %d\n”, i, j);
  }
}
```

Work/Discuss with your friends

```
-------- (f) ---------
j = 5;
for (i= 0; i < j; i++) ; {
    printf ("i= %d, j= %d\n", i, j);
}

-------- (g) ---------
j = 5;
for (i= 0; i < j; j++) {
    printf ("i= %d, j= %d\n", i, j);
}
```

Time for fun: Quick Test

Note: *In all questions, all variables are pre-declared as* `int`.

Q1: What are the values of `s`, `i`, and `c` after the following statement:

```
for (s=0, i=0, c= 0; i<5; i++) {
   s += i;
   c++;
}
```

A	`s= 10, i= 5, c= 5`
B	`s= 10, i= 5, c= 4`
C	`s= 15, i= 6, c= 6`
D	`none of the above`

Time for fun: Quick Test

Q2: How many lines and numbers are printed by the following segment:

```
for (i=0; i<2; i++) {
  for (j=0; j<3; j++) {
    printf("%d ", i*j);
  }
  printf("\n");
}
```

A	6 lines, 6 numbers
B	3 lines, 12 numbers
C	2 lines, 6 numbers
D	none of the above

Time for fun: Quick Test

Q3: How many lines and numbers are printed by the following segment:

```
for (i=0; i<5; i++) {
  for (j=0; j<4; j++) {
    if ( j >i ) continue;
    printf("%d ", i*j);
  }
  printf("\n");
  if (i==2) break;
}
```

A	5 lines, 20 numbers
B	3 lines, 6 numbers
C	2 lines, 3 numbers
D	none of the above

Time for fun: Quick Test

Q4: Which fragment compute s= $1^2 + 2^2 + ... + n^2$?

A	`i=0;` `while (i<=n) s += i*i;`
B	`for (s=0; n > 0; n--) s += n*n;`
C	`for (i=1; i<=n; i++) s += i*i;`
D	`for (s=0,i=1; i<n; i++) s = s + i*i;`

Lab time: VS & gcc! DoItTogether: ex4.5

Design and implement a program `grapher.c` *that reads integers and draw a simple graph. Assume that all of the values read are between 1 and 70. Example:*

```
H: grapher
Enter integers between 1 and 70 inclusive:
3  7  11
  3 |***
  7 |*******
 11 |***********
```

Lab: exercises in grok C04

```
implement("ex 4.9 in grok Week 4");
implement("ex 4.4 in grok Week 4");
for (i=0; i<4 && having_time(); i++) {
  implement("next exercise from grok's W4X");
  if ("the exercise toooo hard") {
    explore("next item in github.com/anhvir/c205");
  }
}
```

Remember:

- Discuss with your classmates
- Ask Anh questions and/or tell him some exciting things

The Coming Quiz 1:

- Do the practice test as many times as possible (you'll have different questions)
- Bring questions to the next week workshop
- Make sure about the Quiz Time and don't forget!

4.9

USE LINK
https://people.eng.unimelb.edu.au/ammoffat/ppsaa/c/isprime.c
(you can Google "Alistair isprime.c") for copying Figure 4.10 page 55

H:> nextprime
Enter an integer value: 8
The next prime is : 11

---------- Figure 4.10 page 55 ----------

```
/* a value for n has been read */
isprime = 1;
for (divisor=2; divisor*divisor <= n; divisor++) {
  if (n % divisor == 0) {
    isprime = 0;
    break;
  }
}
if  (isprime == 1) {
  printf ("%d is a prime number\n",n);
} else {
  printf("%d = %d * %d\n",n, divisor, divisor);
}
```