COMP20005 Workshop Week 4

1 **LAB**	Q&A: loops; Discuss **Ex 4.01** and **4.02** Do Together: **Ex 4.05** Discuss: **Ex 4.09** min requirement: finish **4.05**, **4.09** Implement in the order: **4.04 4.11**
Past & Future	**Lectures W3:** everything about loops; **getchar()**, **putchar()**; **Quiz 1** (Tuesday Week 6): covers chapters 1—5 (ie. till the end of functions) *Thursday class*: Try Sample Tests *before* Workshop Week 5 (if available)

The for ... loop

How to compute:

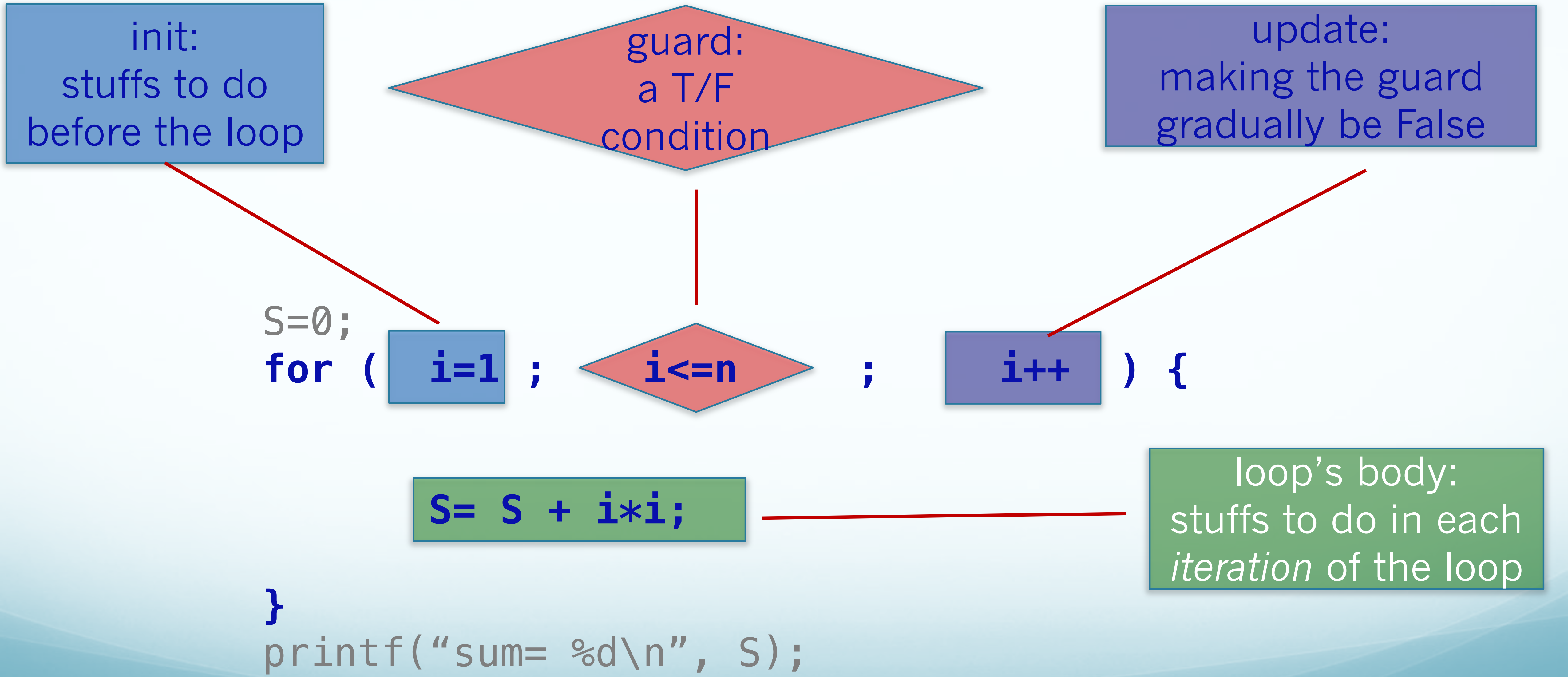
$$S= 1^2 + 2^2 + ... + 5^2$$

How about:

$$S= 1^2 + 2^2 + ... + 100^2$$

$$S= 1^2 + 2^2 + ... + n^2$$

The for . . . loop



init:
stuffs to do
before the loop

guard:
a T/F
condition

update:
making the guard
gradually be False

loop’s body:
stuffs to do in each *iteration* of the loop

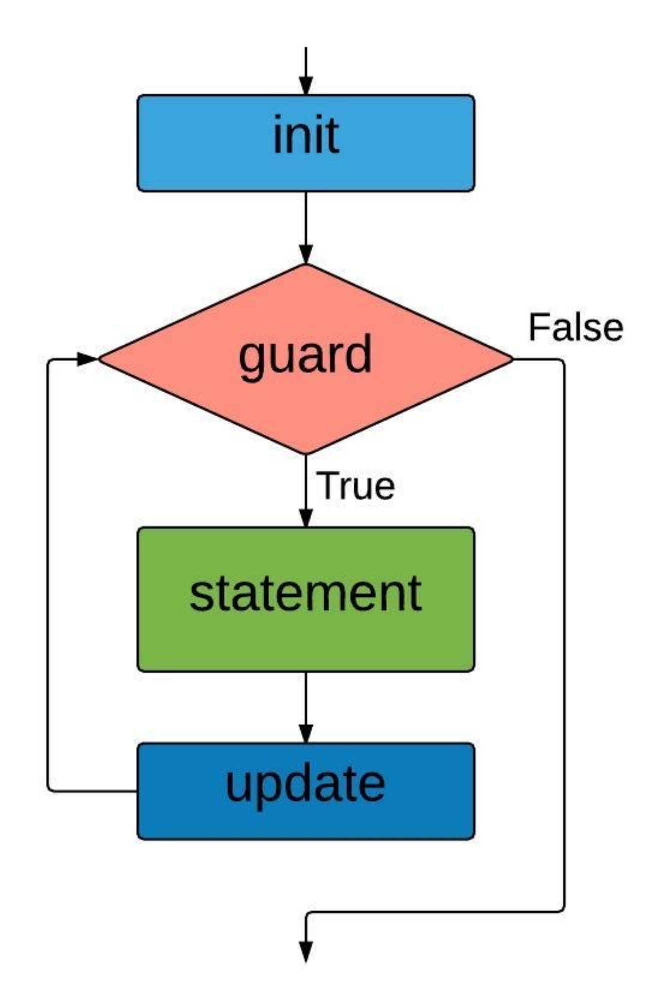
```
S=0;
for ( i=1 ; i<=n ; i++ ) {
    S= S + i*i;
}
printf(“sum= %d\n”, S);
```

can be empty, empty means 1

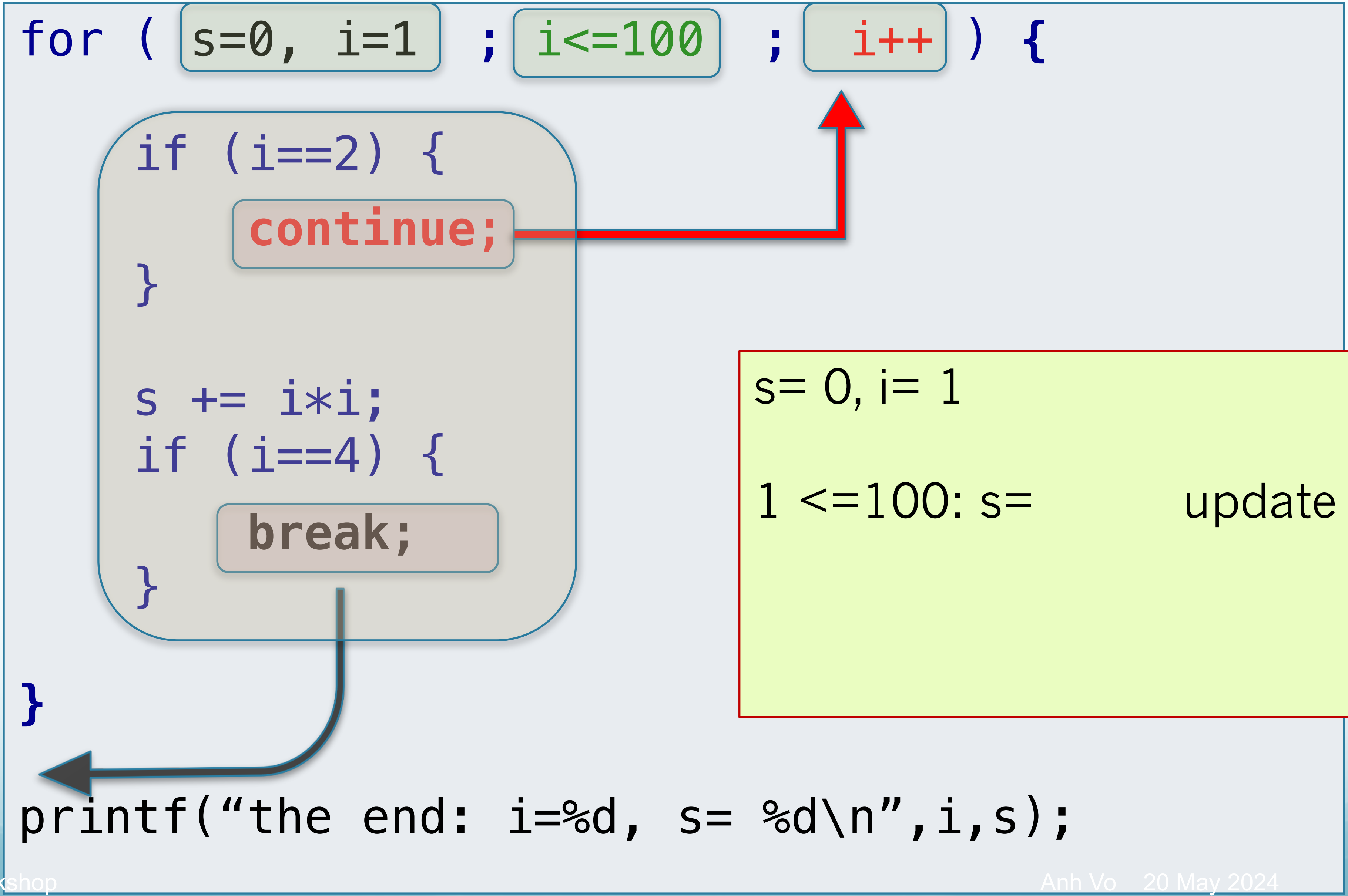
```
for ( init ; guard     ; update ) {

        statement;...

}
```

All the boxes can be empty! The must-be parts are:

```
for (  ;  ;  )
```



continue (go to <update>) & break (exit loop)



```
for ( s=0, i=1 ; i<=100 ; i++ ) {

    if (i==2) {
        continue;
    }

    s += i*i;
    if (i==4) {
        break;
    }

}

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

s= 0, i= 1

1 <=100: s= update i=

the while ... loop

- could be considered as a special case of the for loop, when init and update are both empty

```
while ( condition ) {
      statements
}
```

⟺

```
for ( ; condition ; ) {
      statements
}
```

```
init
while ( condition ) {
   statements
   update
}
```

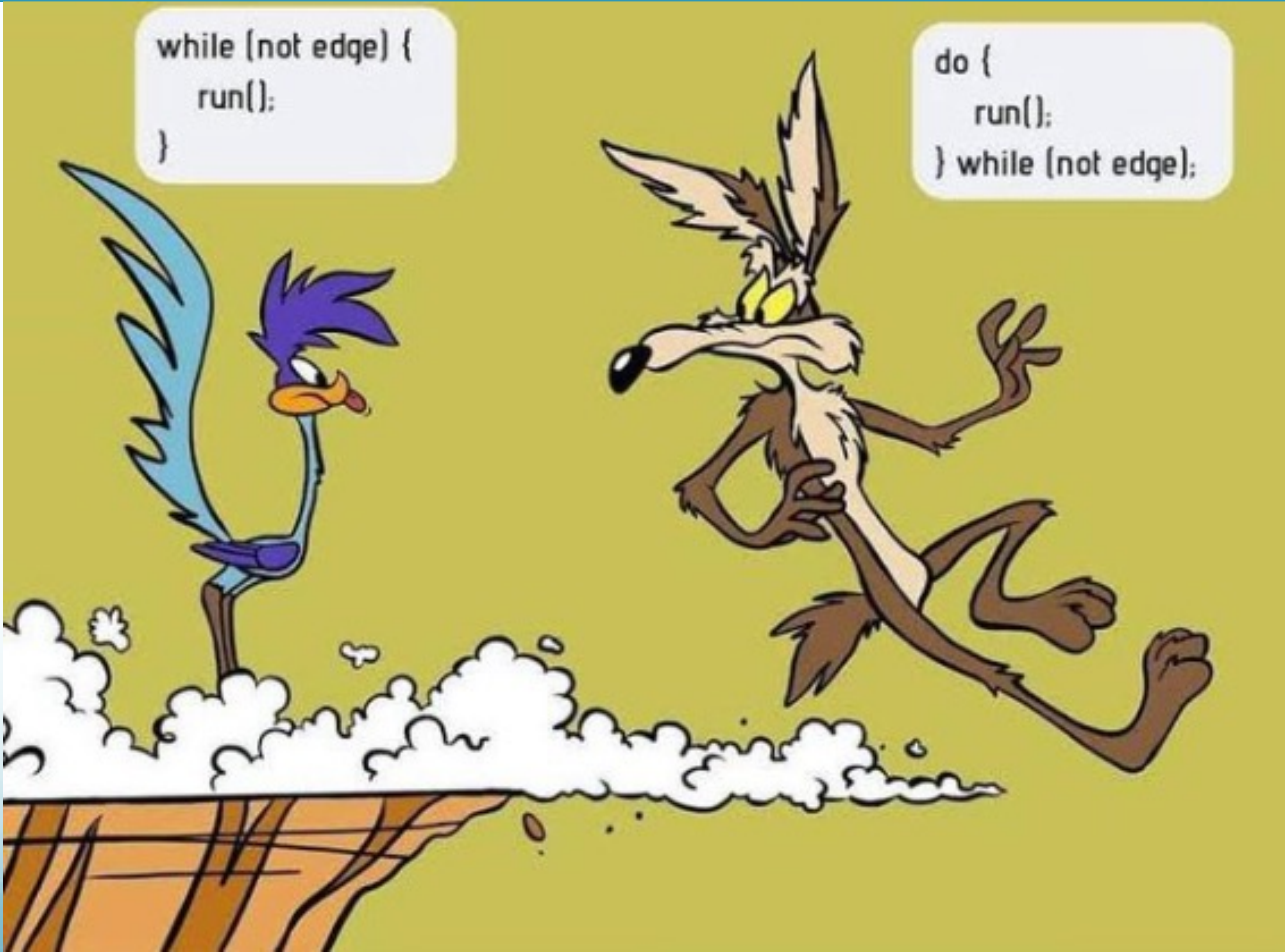
⟺

```
for (init; condition ; update) {
      statements
}
```

Loops: while and do…while

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

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



```
while (not edge) {
    run();
}
```

```
do {
    run();
} while (not edge);
```

Ex 4.02

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;
}
```

Writing Loops (use Playground or any exercise)

Write the code segment to compute:

sum of integers read from the keyboard, stop when input symbol is not part of an integer

Ex 4.01 a–c)

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`.

```
for (i=0; i<20; i=i+3) {
   printf("%2d\n", i);
}
```

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

```
sum = 0;
for (i = 1; i < 10; i++) {
   sum = sum + i;
   printf("S(%2d) = %2d\n", i, sum);
}
```

Ex 4.01 d) (changed)

```
for (i= 0; i < 4; i++) {

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

}
```

```
i= , j=
```

Ex 4.01 e) (changed)

```
for (i= 2; i < 5; i++) {
 for  (j= i+1; j < 8; j += 3) {
  if (i+j == 7) {
   break;
  }
  printf (“i= %d, j= %d\n”, i, j);
 }
}
```

```
i= , j=
```

Ex 4.01 f)

```c
j = 5;
for (i= 0; i < j; i++) ; {
    printf (“i= %d, j= %d\n”, i, j);
}
```

```
i= , j=
```

How many lines printed out?

Ex 4.01 g)

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

```
i= , j=
```

How many lines printed out?

Do-Together in grok: `Ex 4.05`

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:*

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

How to approach Ex 4.09

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

min requirement: finish **`4.05`**, **`4.09`**

others: **`4.04 4.03 4.06-4.07 4.x1 4.x2 ...`**

Remember:

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

At home: Preparing for the Mid-Semester Quiz on Tuesday Week 6 by

- Reviewing lectures
- Practicing programming: at least do all grok exercises of C01-C05
- Trying the practice test with timing. Note: you can do a few times, and you'll probably have different questions each time.
- Bringing questions to the next week workshop
- Making sure about the Quiz: Where, When, What Rules

Wrap Up

```
for (<I> ;  <?> ;  <U>) {
      //do one iteration
}
```

```
while (<?>) {
      //do one iteration
}
```