ECE 220: Computer Systems & Programming

Lecture 6: Control Structures & Basic I/O

Thomas Moon

February 6, 2024

Mock quiz	01/30 - 02/01	CBTF	Short Survey LC-3 practice (HW0)
Quiz 1	02/05 - 02/07	CBTF	LC-3 Programming (up to and including Lecture 4/Lab 1 /MP 1) See Lab1 and Lab2 programming exercise
			An LC-3 web simulator will be available during Quiz 1 - you should NOT use it for your MPs.

Exam	Date & Time	Location	Topic	Practice Questions	Conflict Exam Information
Midterm 1	Thursday 02/15 at 7.00pm - 8.20pm	See below	Lecture 1 to Lecture 06 Associated book chapters, labs,and MPs. (Programming & Concept)	Past exams Worksheets	Submit request **Deadline: 02/11**

Some good questions

- Can a function in C/C++ return multiple outputs?
  - No! Only one return value.
  - However, a function in c can modify multiple variables by Pointers!

Practice

```
int a = 6, b = 9;

Expression      Value of Expression

a | b
a || b
a & b
a && b
!(a + b)
a % b
b / a
a == b
a = b
a = b = 5
++a + b--
a→6, b→4
```

Practice

```
int a = 6, b = 9;
```

Expression	Value of Expression
a | b	0b0110 | 0b1001 = 0b1111 = 15
a || b	1
a & b	0b0110 & 0b1001 = 0b0 = 0
a && b	1
!(a + b)	!(15) = !(true) = 0
a % b	6
b / a	9/6 = 1
a == b	0
a = b	9
a = b = 5	5
++a + b--	6 + 5 = 11

a→6, b→4

Control Structures

- Conditional

  Making a decision about which code to execute, based on evaluated expression

  - `if`
  - `if-else`
  - `switch`

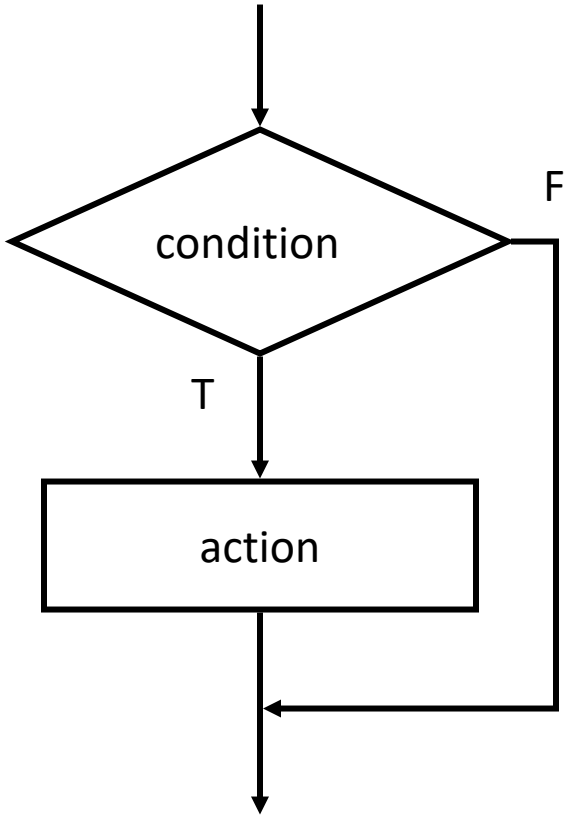
- Iteration

  Executing code multiple times, ending based on evaluated expression

  - `while`
  - `for`
  - `do-while`

if

```
if (condition)
  action;
```



- Condition: C expression, which evaluates to TRUE (non-zero) or FALSE(zero)
- Action: C statement, which will be executed if condition is TRUE

Example if statements

```
if (x <= 10)
    y = x * x + 5;
```

same

```
if (x <= 10){
    y = x * x + 5;
}
```

```
if (x <= 10){
    y = x * x + 5;
    z = (2*y)/3;
}
```

NOT same

```
if (x <= 10)
    y = x * x + 5;
    z = (2*y)/3;
```

C is NOT “indentation-based” language like Python

Example if statements (continued)

```
if (0 <= age && age <= 11){
    kids += 1;
}
```

```
if (0 <= age <= 11)
-> Wrong!
```

```
if (month == 4 || month == 6 ||
month == 9 || month == 11){
    printf("The month has 30 days.\n");
}
```

```
if (x = 2){
    y = 5;
}
```

The condition is Always true!

-> Common programming error (= instead of ==) not caught by compiler because it's syntactically correct.

Is `if` enough?

```c
int num1, num2, res;
char op;
printf("Enter an equation (+,-,*):");
scanf("%d %c %d", &num1, &op, &num2);

if(op == '+'){
    res = num1 + num2;
}
if(op == '-'){
    res = num1 - num2;
}
if(op == '*'){
    res = num1 * num2;
}
printf("Result: %d\n", res);
return 1;
```

```
Enter an equation (+,-,*):4 - 5
Result: -1
```

What if invalid input?

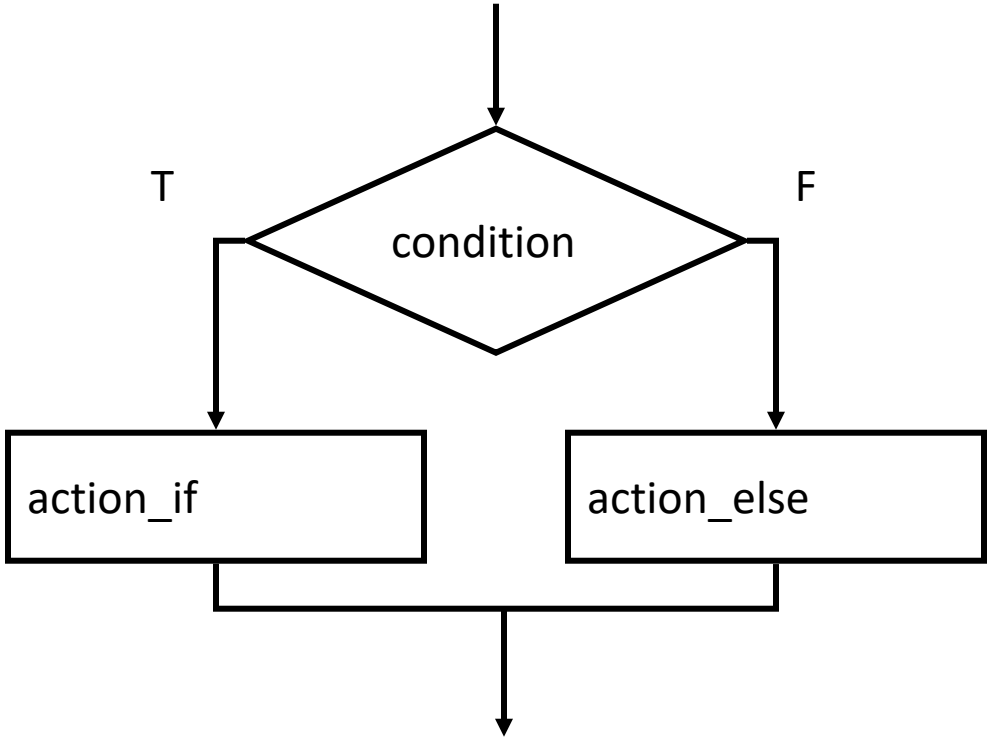
```
Enter an equation (+,-,*):5 / 5
Result: 0
```

→"if" itself cannot cover the rest cases
→if-statements are independent

if-else

```
if (condition)
  action_if;
else
  action_else;
```

```
if (x < 0){
   x = -x;
}
else{
   x = x * 2;
}
```



- Else allows choice between two mutual-exclusive actions.

Chaining `if/else`

```c
int num1, num2, res;
char op;
printf("Enter an equation (+,-,*):");
scanf("%d %c %d", &num1, &op, &num2);

if(op == '+'){
    res = num1 + num2;
}
else if(op == '-'){
    res = num1 - num2;
}
else if(op == '*'){
    res = num1 * num2;
}
else{
    printf("Invalid operator.\n");
    return 0;
}
printf("Result: %d\n", res);
return 1;
```

```
Enter an equation (+,-,*):4 / 3
Invalid operator.

Enter an equation (+,-,*):3 4 5
Invalid operator.
```

Floating Number Comparison (Caution)

```
float myFloat = 3.14;

if(myFloat == 3.14)
    printf("My float is PI.\n");
else
    printf("My float is not PI.\n");
```

```
My float is not PI.
```

```
double myDouble = 3.14;

if(myDouble == 3.14)
    printf("My double is PI.\n");
else
    printf("My double is not PI.\n");
```

```
My double is PI.
```

```
printf("%d, %d, %d\n", sizeof(3.14), sizeof(3.14f),  sizeof(myFloat));
```

```
8, 4, 4
```

String Comparison (More on Lecture 12)

```
char str1[10] = "ABC";
char str2[10] = "ABC";
if(str1 == str2)
    printf("Both \"ABC\"\n");
else
    printf("Not same\n");
```

```
Not same
```

```
printf("str1=%d, str2=%d\n", str1, str2);
```

```
str1=599334208, str2=599334192
```

→ *They hold the address of the first character. (different address each run!)*

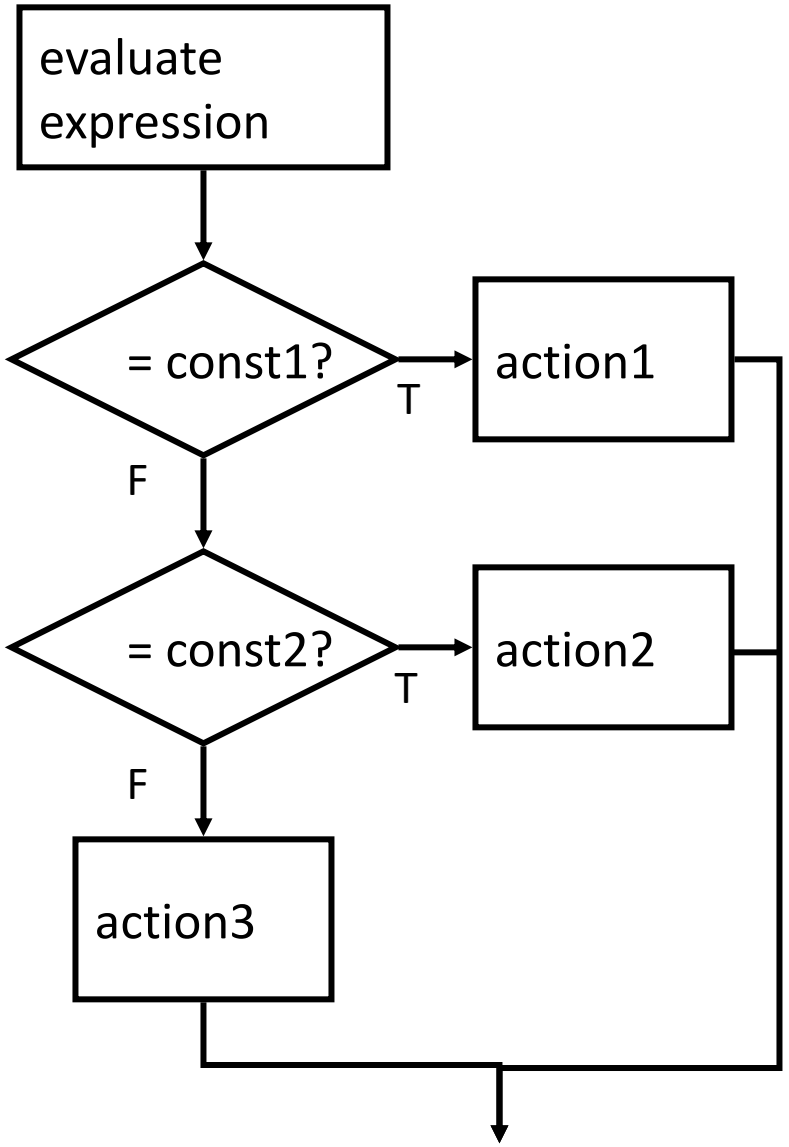
```
#include <string.h>
. . .

if( strcmp(str1, str2) == 0 )
    printf("Both \"ABC\"\n");
```

```
Both "ABC"
```

switch

```
switch (expression) {
case const1:
  action1; break;
case const2:
  action2; break;
default:
  action3;
}
```



- Alternative to long if-else chain.
- If break is not used,
  then cases ”fall through” to the next

Break Example

```
a = 1;
switch(a){
    case 1:
        printf("A");
        break;
    case 2:
        printf("B");
        break;
    default:
        printf("C");
        break;
}
```

A

```
a = 1;
switch(a){
    case 1:
        printf("A");
    case 2:
        printf("B");
    default:
        printf("C");
}
```

ABC

```
a = 2;
switch(a){
    case 1:
        printf("A");
    case 2:
        printf("B");
    default:
        printf("C");
}
```

BC

Example

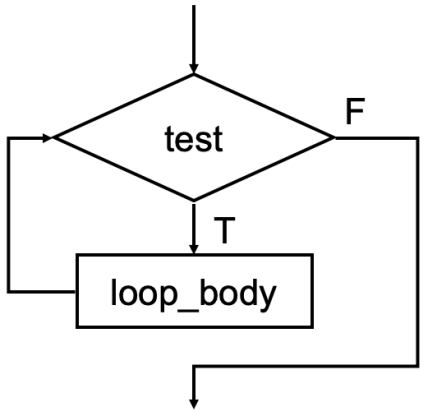
```
/* same as month example for if-else */
switch (month) {
  case 4:
  case 6:
  case 9:
  case 11:
    printf("Month has 30 days.\n");
    break;
  case 1:
  case 3:
  /* some cases omitted for brevity...*/
    printf("Month has 31 days.\n");
    break;
  case 2:
    printf("Month has 28 or 29 days.\n");
    break;
  default:
    printf("Don't know that month.\n");
}
```

Let's use `switch`

```
int num1, num2, res;
char op;
printf("Enter an equation (+,-,*):");
scanf("%d %c %d", &num1, &op, &num2);
```

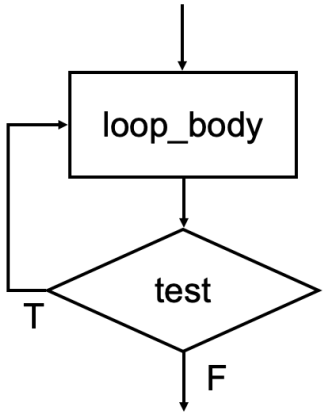
while

```
while (test){
  loop_body;
}
```



- Test is evaluated before executing loop body

```
do{
  loop_body;
}while (test);
```



- Test is evaluated after executing loop body

Example

```
int num = 3;
while(num > 0){
    printf("%d", num);
    num--;
}
```

3 2 1

How many loops?

Non-zero → True
Zero → False

```
while(0){
    printf("A");
}
```

while(0.0) ?

0

```
while(1){
    printf("A");
}
```

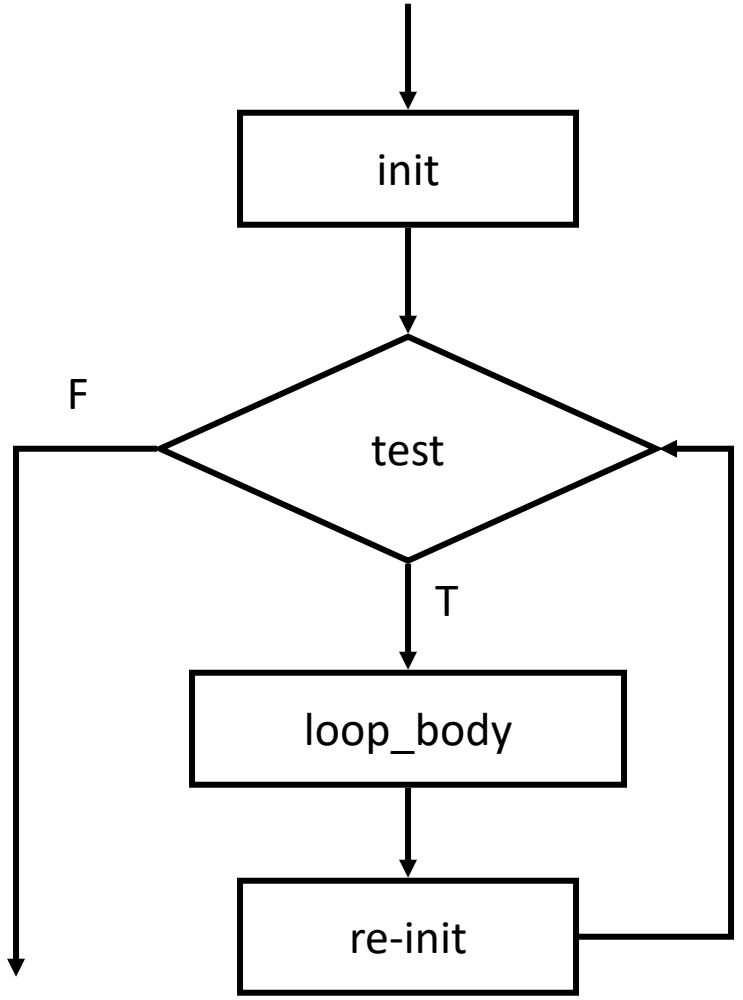
Infinite

```
while(1.5){
    printf("A");
}
```

Infinite

for

```
for (init; end-test; re-init)
   statement
```



while vs do-while vs for

print out
0123456789

while

```
x = 0;
   while( x < 10  ){
       printf("%d", x);
       x++;
   }
```

do-while

```
x = 0;
  do{
      printf("%d", x);
      x++;
  }while( x < 10);
```

for

```
for( x = 0 ; x < 10 ; x++ ){
      printf("%d", x);
  }
```

```
for( x = 0 ; x < 10 ; ++x ){
      printf("%d", x);
  }
```

break vs continue

- break
  - used only in switch or iteration statement
  - used to exit a loop before terminating condition occurs
- continue
  - used only in iteration statement
  - end the current iteration and start the next

```
for (i = 0; i < 10; i++){
    if(i == 5)
        break;
    printf("%d ",i);
}
```

```
0 1 2 3 4
```

```
for (i = 0; i < 10; i++){
    if(i == 5)
        continue;
    printf("%d ",i);
}
```

```
0 1 2 3 4 6 7 8 9
```

Problem: Print nxn Identity Matrix

- 3-by-3 identity matrix :

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

- Can we stop printing after the second “1” on the main diagonal such as

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & \end{matrix}$$

Midterm1 (Lec1-6) Review:

- 4 problems, 80 min
  - 2 programming questions (LC3, fill-in-the-blank questions)
  - 1 debugging question (LC3)
  - 1 concept question (multiple sub-qeustions, LC3+C)
- ASCII and LC-3 instruction table will be given.
- No cheat sheet, no calculator

Midterm1 Review:

- Basic concept on LC-3
  - memory, processing unit, control unit, input/output
- Memory mapped I/O
  - KBDR, KBSR, DDR, DSR
  - Basic input/output routine by polling
- TRAP
  - TRAP mechanism operation (TVT, TRAP service routine, …)
- Subroutine
  - How to write a subroutine (callee/caller-save, RET, R7, nested subroutine, …)
- Stack
  - PUSH, POP, TOS(R6)
- C
  - operators, if/else, while/for, etc
  - printf/scanf

- **Review your MP1/2/3, worksheets**
- Be familiar with LC-3 instructions (e.g. ST/LD family, Branch nzp?)