Anil Neerukonda Institute of Technology & Sciences
Department of Computer Science & Engineering (AI & ML, DS)

CSE117 Problem Solving with C

Handout - Lab Session - 4

Multiway Selection & Repetetion

Objective:

- To write C programs that use multi-way selection: switch and else if.
- To be able to use bitwise logical operators in C programs.
- To be able to use bitwise shift operators in C programs.
- To write C programs to create and use masks to manipulate bits.
- To write C programs using while and for statements.

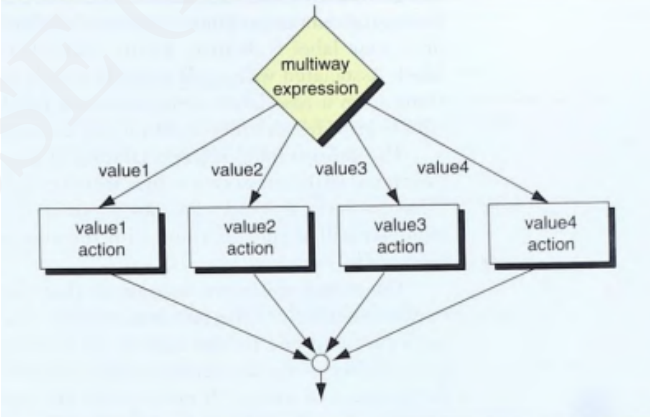
Pre-Lab: Go through the concepts of multi-way selection, bitwise operators, and repetition. Write the algorithm and flowcharts for all the exercise problems given in this handout.

During Lab: Solve all the exercise problems. You should work on the additional set of programs only after completing this week's tasks.

Post Lab: Take the quiz.

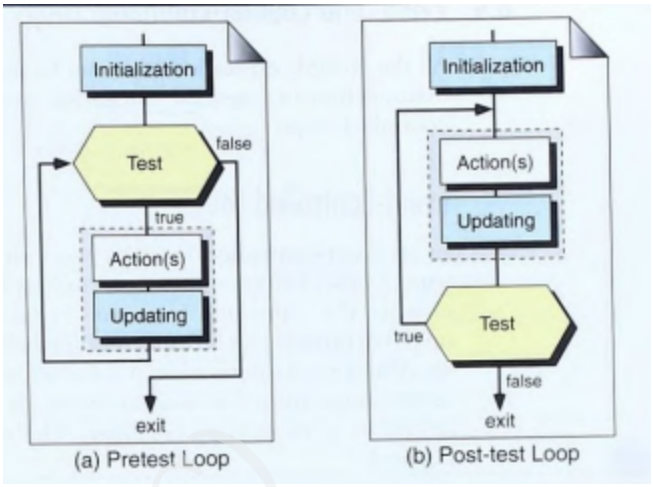
Multiway Selection

Multiway



** Refer to the textbook for more information. (Chapter 5, A structured programming approach using C. Behrouz A. Forouzan & Richard F. Gilberg)

Repetition



The real power of computers is in their ability to repeat an operation ora series of operations many times. To control the loop, we need a condition to determine if more processing is needed. This condition is called a ***loop controlled expression.***

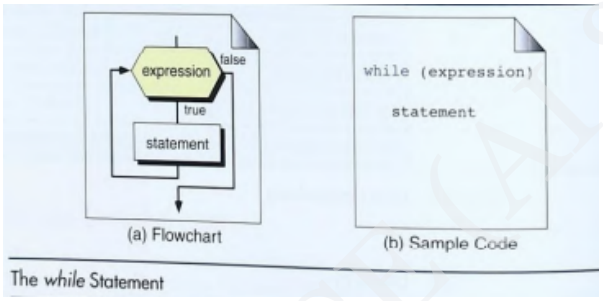
In a ***pretest loop***, in each iteration, we check the condition first. If it is true, we iterate once more; otherwise, we exit the loop. In a ***post-test loop***, in each iteration, we do the processing. Then we check the condition. It it is true, we start a new iteration; otherwise, we exit the loop.

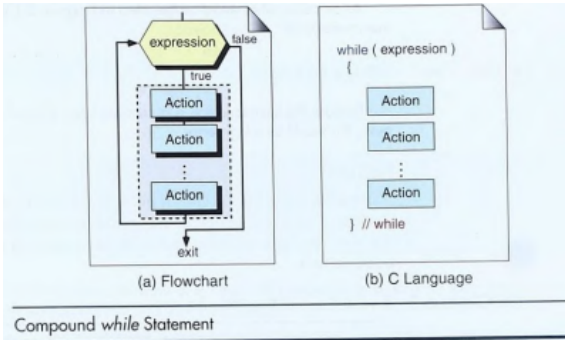
If we know exactly the number of times the loop body must be repeated, we use a ***counter-controlled loop***; if an event must occur to terminate a loop, use an ***event-controlled loop***.

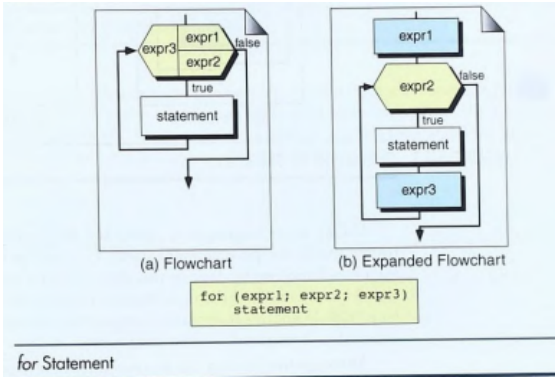
C has three loop statements: while, for, and do...while.

The ***while loop*** is a pretest loop. It can be used for a counter-controlled or event-controlled loop, but it is usually used only for event control. The ***for loop*** is a pretest loop. It can be used for both counter-controlled and event-controlled loops, but it is used mostly in the first case.

The syntax of while loop and for loop are given below.







** Refer to the textbook for more information. (Chapter 6, A structured programming approach using C. Behrouz A. Forouzan & Richard F. Gilberg)

Lab Exercises

Design algorithm, flow chart, and program using the data requirements given for the exercise problems and try all the test cases.

Exercise 1: **Set, Reset or Toggle a specified bit in a 16-bit unsigned integer.**

Read two numbers from the user at the keyboard. The first number is the integer to be manipulated, and the second number is the bit location that is to be manipulated.
Display the menu to the user and ask them to choose an option.

Sample Test Cases	Input	Output
Test Case 1	Enter the number (16-bit unsigned integer) : 0. Enter the bit to be manipulated: 5 1. Set a bit 2. Reset a bit 3. Toggle a bit >> 1	16
Test Case 2	Enter the number (16-bit unsigned integer): 28 Enter the bit to be manipulated: 5 1. Set a bit 2. Reset a bit 3. Toggle a bit >> 2	12
Test Case 3	Enter the number (16-bit unsigned integer): 28 Enter the bit to be manipulated: 5 1. Set a bit 2. Reset a bit 3. Toggle a bit >> 3	12

Exercise 2: **Times Table Generator**

Read non-negative integer, n, count from the user. Print the multiplication table or n from 1 to count.

Sample Input	Sample Output
n : 3 count: 10	3 x 1 = 3 3 x 2 = 6 3 x 3 = 9 ….. ….. 3 x 10 = 30

Exercise 3: **Factorial of a given number.**

Read a non-negative integer, n, from the user and print its factorial.

Sample Test Cases	Input	Output
Test Case 1	5	120
Test Case 2	20	2432902008176640000
Test Case 3	-3	Invalid input.

Exercise 4: **Fibonacci Series**

Read a non-negative integer, n, from the user and print n Fibonacci terms.

Sample Test Cases	Input	Output
Test Case 1	6	0 1 1 2 3 5
Test Case 2	1	0
Test Case 3	-6	Invalid input
Test Case 4	0	Invalid input

Exercise 5: **Average of Positive numbers**

Given a positive integer, n, read n floating-point numbers. As the numbers are read, the program should calculate the sum and average of the positive numbers.

Sample Test Cases	Input	Output
Test Case 1	n : 5 1 2.3 -1.2 3.1 -2	Sum : 6.40 Average: 2.13
Test Case 2	n : 3 0 -1 2	Sum: 2.00 Average: 1.00

Exercise 6: Print Negative Numbers and Find **Average of Positive numbers.**

Given a positive integer, n, read n integers. As the numbers are read, the program should calculate the average of the positive numbers and print the negative numbers.

Sample Test Cases	Input	Output
Test Case 1	n: 5 1 2.3 -1.2 3.1 -2	-1.2 -2 Sum: 6.40 Average: 2.13
Test Case 2	n : 3 0 -1 2	-1 Sum: 2.00 Average: 1.00

Exercise 7: **Prime Sum**

Given a positive integer, n, calculate the sum of all prime numbers between 1 and n (inclusive).

Sample Test Cases	Input	Output
Test Case 1	5	10
Test Case 2	10	17