

APCO 1P93 - Lab 3

Due Date: End of Wednesday Feb 29th Lab

To receive credit for this lab, show your programs to the TA before you leave the lab.

PART 1 - Logical Operators

There are three logical operators in C - and (&&), or (||), not (!). We want to see the results by using if statements to test all of the possible states of the operands.

The following is the code in order to figure out a and b.

```
int a, b;

a = 5, b = 10;

if ((a == 5) && (b == 10)) {
    printf("true && true is true\n");
}
else {
    printf("true && true is false\n");
}

if ((a == 5) && (b == 8)) {
    printf("true && false is true\n");
}
else {
    printf("true && false is false\n");
}
```

Compile this code, follow the pattern and find the results for A - J.

- A) true && true _____
- B) true && false _____
- C) false && true _____
- D) false && false _____
- E) true || true _____
- F) true || false _____
- G) false || true _____
- H) false || false _____
- I) ! true _____
- J) ! false _____

PART 2 - Switch Statement

The switch statement is used in C to make a decision between many alternatives.

```
int score, mark;

printf("Enter the test score (0-100): ");

scanf("%d", &score);

mark = score / 10;
printf("\n\n");

switch (mark) {
    case 10:
    case 9:
        printf("your mark is an A\n\n");
        break;
    case 8:
        printf("your mark is a B\n\n");
        break;
    case 7:
        printf("your mark is a C\n\n");
        break;
    case 6:
        printf("your mark is a D\n\n");
        break;
    default:
        printf("your mark is an F\n\n");
} // switch
```

Compile the above and find the results for the different score values below. After step E, remove all of the break statements, recompile and find the results for F - H.

- A) 67 _____
- B) 89 _____
- C) 100 _____
- D) 39 _____
- E) 96 _____
- F) 67 _____
- G) 47 _____
- H) 87 _____

PART 3 - Else-if structure

The else-if structure is used to make a multiway decision, like the switch statement, but can be used with values that are not integrals. For instance, based on what we learned in part 1, only one of the 4 conditions below will print out. Compile the code and note the syntax structure of the else-if statements. Change the values of A and B and note which statement prints out.

```
int a, b;
a = 5, b = 8;
```

```

if ((a == 5) && (b == 10)) {
    printf("both conditions are true\n");
}
else if ((a == 5) && (b == 8)) {
    printf("only first condition is true\n");
}
else if ((a == 3) && (b == 10)) {
    printf("only second condition is true\n");
}
else {
    printf("both conditions are false");
}
}

```

Using the above template, code the following problem. Prompt for a number between 0 and 100. Based on that number, print out one of the following four statements:

- number is between 0 and 50, number is an odd number
- number is between 0 and 50, number is an even number
- number is between 51 and 100, number is an odd number
- number is between 51 and 100, number is an even number

PART 4 - for loop, while loop and do...while loop

The following 3 code segments all achieve the same goal. Copy this code and run it within a C program. Change some of the values of the variables i and j and see what happens.

a) for loop

```

int i, j;
j = 5;

printf("\n*****part a - start of for loop\n");

for (i = 1; i <= j; i++) {
    printf("the value of i... %d\n",i);
}

printf("*****part a - end of for loop\n");
printf("value afterwards of i is %d\n\n",i);

```

b) do...while loop

```

int i, j;
i = 1;
j = 5;

printf("\n*****part b - start of do...while loop\n");

do {
    printf("the value of i... %d\n",i);
    i++;
}

```

```

    } while (i <= j);

    printf("*****part b - end of do...while loop\n");
    printf("value afterwards of i is %d\n\n",i);

```

c) **while loop**

```

int i, j;

i = 1;
j = 5;

printf("\n*****part c - start of while loop\n");

while (i <= j) {
    printf("the value of i... %d\n",i);
    i++;
};

printf("*****part c - end of while loop\n");
printf("value afterwards of i is %d\n\n",i);

```

PART 5 - *for* loop...starting at something other than 0

Copy the following code into your C program and run it. This code will total the numbers between 100 and 200. After this, add in a counter inside the for loop to keep track of how many numbers you have added to make sure that you have added 100 numbers.

d) **for loop**

```

int total = 0;
int ctr;

printf("\n*****part d - start of for loop\n");

for (ctr = 100; ctr <= 200; ctr++)
    total += ctr;

printf("the total of then numbers between 100 and 200 is
%d\n",total);
printf("*****part d - end of for loop\n");

```

e) Convert part d so that it is a *while* loop instead of a *for* loop

PART 6 - *for* loop...incrementing by a value other than 1

Copy the following code into your C program and run it. This code will print all of the even numbers under 20.

f) **for loop**

```

int num;

printf("\n*****part f - start of for loop\n");

for (num = 2; num <= 20; num = num + 2)
    printf("%d ", num);

printf("\n*****part f - end of for loop\n");

```

g) Using part f as an example, print all of the odd numbers under 30, in reverse order (ie 29 to 1).

PART 7 - nested for loop

Copy the following code into your C program and run it. Try to follow the logic of the nested loops.

h) **nested for loop**

```

int outer, inner;

printf("\n*****part h - start of for loop\n");

for (outer = 9; outer >= 1; outer--) {

    for (inner = 1; inner <= outer; inner++) {
        printf("%d", inner);
    }
    printf("\n");

}

printf("\n*****part h - end of for loop\n");

```

i) Using part h as an example, write the code that will produce this as output:

```

1
13
135
1357
13579

```

PART 8 - for loop...printing out the counter

j) Write the code that will prompt the user for how old they are. For every year they have been alive, write to the screen the line 'Happy Birthday!\n'

PART 9 - for loop...using the break statement

Copy the following code into your C program and run it. This code will prompt the user for 5 ratings. However, if it is not a valid rating (between 1 and 5), the user is displayed an error message and asked to try later.

k) for loop with break

```

int ctr;
float average, rating, total;

```

```

printf("\n*****part k - start of for loop\n");

for (ctr = 0; ctr < 5; ctr++) {

    printf("Please enter a rating (1-5): ");
    scanf("%f",&rating);

    if ((rating >= 1) && (rating <= 5)) {
        total += rating;
    }

    else {
        printf("You have entered an invalid
rating...please come back later\n\n");
        break;
    }
    printf("\n");

}

average = total / ctr;
printf("\n\nThe average rating is %f\n\n",average);

```

There is a slight problem with the above code though. What happens if the user enters an invalid rating for their first rating? What average is printed out? Add an if statement to prevent the average from being calculated and being displayed if there are no valid ratings.

URL of this document: <http://www.cosc.brocku.ca/Offerings/APCO/1P93/labs/lab3.html>
Revised: 02/07/2012
Instructor: James Bell