HW04 Problem Set

CS-3160

Assignment

HW04: Sebesta Problems 7.(7,8,9,10,11,12,13,18,20), Programming Exercises 7.(1,8)
HANDWRITTEN – Due at beginning of class on due date.
PROGRAM – Due at midnight, via Blackboard, at midnight on day prior to due date.

Handwritten Portion

7. Describe a situation in which the add operator in a programming language would not be commutative.
8. Describe a situation in which the add operator in a programming language would not be associative.
9. Assume the following rules of associativity and precedence for expressions:

Precedence	*Highest*	*, /, **not**
		+, –, &, **mod**
		– (unary)
		=, /=, < , <=, >=, >
		and
	Lowest	**or**, **xor**
Associativity	*Left to right*	

Show the order of evaluation of the following expressions by parenthesizing all subexpressions and placing a superscript on the right parenthesis to indicate order. For example, for the expression

```
a + b * c + d
```

the order of evaluation would be represented as

$((a + (b * c)^{1})^{2} + d)^{3}$

a. `a * b - 1 + c`
b. `a * (b - 1) / c` **`mod`** `d`
c. `(a - b) / c & (d * e / a - 3)`
d. `-a` **`or`** `c = d` **`and`** `e`
e. `a > b` **`xor`** `c` **`or`** `d <= 17`
f. `-a + b`

HW04 Problem Set

CS-3160

10. Show the order of evaluation of the expressions of Problem 9, assuming that there are no precedence rules and all operators associate right to left.

11. Write a BNF description of the precedence and associativity rules defined for the expressions in Problem 9. Assume the only operands are the names `a,b,c,d,` and `e`.

12. Using the grammar of Problem 11, draw parse trees for the expressions of Problem 9.

13. Let the function fun be defined as

```
int fun(int *k) {
 *k += 4;
 return 3 * (*k) - 1;
 }
```

Suppose fun is used in a program as follows:

```
void main() {
  int i = 10, j = 10, sum1, sum2;
  sum1 = (i / 2) + fun(&i);
  sum2 = fun(&j) + (j / 2);
 }
```

What are the values of `sum1` and `sum2`

a. if the operands in the expressions are evaluated left to right?

b. if the operands in the expressions are evaluated right to left?

18. Should an optimizing compiler for C or C++ be allowed to change the order of subexpressions in a Boolean expression? Why or why not?

HW04 Problem Set

CS-3160

20. Consider the following C program:

```
int fun(int *i) {
  *i += 5;
  return 4;
 }
void main() {
  int x = 3;
  x = x + fun(&x);
 }
```

What is the value of x after the assignment statement in main, assuming

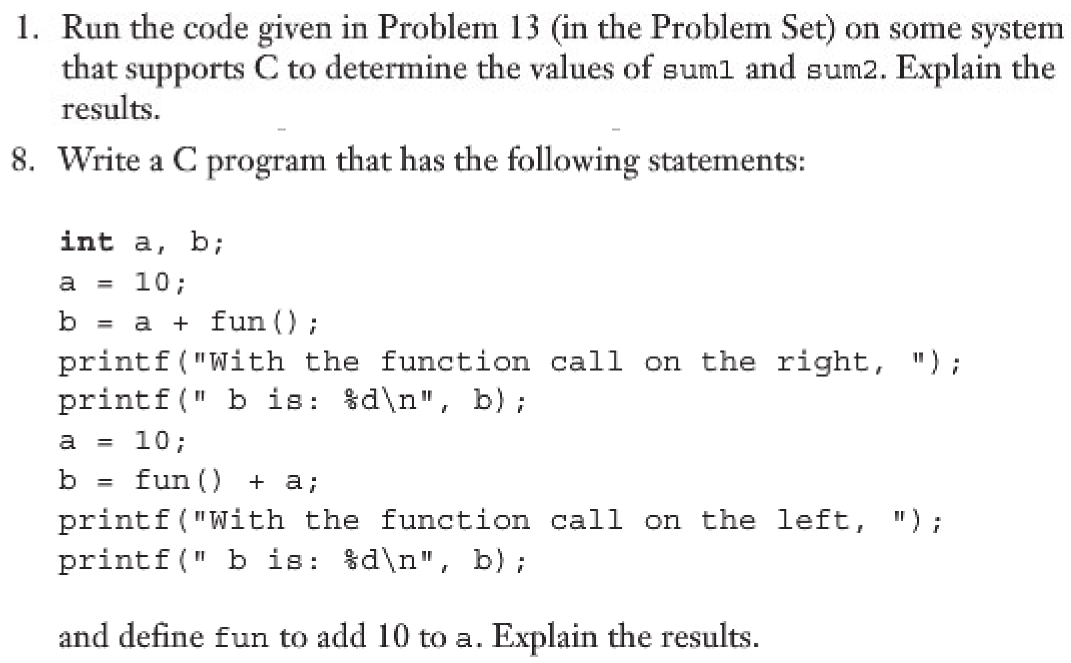
a. operands are evaluated left to right.

b. operands are evaluated right to left.

HW04 Problem Set

CS-3160

Programming Portion



1. Run the code given in Problem 13 (in the Problem Set) on some system that supports C to determine the values of `sum1` and `sum2`. Explain the results.

8. Write a C program that has the following statements:

```
int a, b;
a = 10;
b = a + fun();
printf("With the function call on the right, ");
printf(" b is: %d\n", b);
a = 10;
b = fun() + a;
printf("With the function call on the left, ");
printf(" b is: %d\n", b);
```

and define `fun` to add 10 to `a`. Explain the results.

NOTES

In Programming Exercise 7.8, the function fun() should change the value stored in the variable **a** by increasing it by ten. Thus **a** needs to be accessible to fun(). Easiest way: make **a** and **b** global variables.

Grading Rubric

The assignment is worth 25 pts (as a whole) and the score will be recorded as a percentage of that amount.

Problem	7	8	9	10	11	12	13	18	20
Points	1	1	3	3	2	3	2	2	2

Programming Problems: 7.1 - 2pts; 7.8 – 4pts

10% Physical Format
50% Answers correct (and supported by work)
40% Effort evidenced by the submitted work