

WEEK-END ASSIGNMENT-01

Operating Systems Workshop (CSE 3541)

Problem Statement:

Experiment with C operators, role of operator precedence, associativity and expressions.

Assignment Objectives:

To become familiar with C operators, expression evaluation as per operator precedence and associativity rule.

Instruction to Students (If any):

Students are required to write his/her own program by avoiding any kind of copy from any sources. Additionally, They must be able to realise the outcome of that question in relevant to systems programming.

Programming/ Output Based Questions:

1. Evaluate the arithmetic expression $a - b/c * d$, where the floating-point variables a , b , c and d have been assigned the values 1.0, 2.0, 3.0 and 4.0. Create a C program to display the value of the expression on standard output device.

Write/paste your code here ▼	Output ▼

2. Which of the following identifiers are (a) C reserved words, (b) standard identifiers, (c) conventionally used as constant macro names, (d) other valid identifiers, and (e) invalid identifiers?

<code>void</code>	<code>MAX_ENTRIES</code>	<code>return</code>	<code>printf</code>	<code>"char"</code>
<code>xyz123</code>	<code>time</code>	<code>part#2</code>	<code>G</code>	<code>Sue's</code>
<code>#insert</code>	<code>this_is_a_long_one</code>		<code>double</code>	<code>__hello__</code>

Answer here ▼				

3. The Pythagorean theorem states that the sum of the squares of the sides of a right triangle is equal to the square of the hypotenuse. For example, if two sides of a right triangle have lengths of 3 and 4, then the hypotenuse must have a length of 5. Together the integers 3, 4, and 5 form a *Pythagorean triple*. There are an infinite number of such triples. Given two positive integers, m and n , where $m > n$, a Pythagorean triple can be generated by the following formulas:

$$side1 = m^2 - n^2$$

$$side2 = 2mn$$

$$hypotenuse = m^2 + n^2$$

The triple ($side1 = 3$, $side2 = 4$, $hypotenuse = 5$) is generated by this formula when $m = 2$ and $n = 1$. Write a program that takes values for m and n as input and displays the values of the Pythagorean triple generated by the formulas above. The values of m and n should be provided from an input file through input redirection.

Write/paste your code here ▼	Output ▼

4. Write C statements to carry out the following steps.
- If **item** is nonzero, then multiply **product** by **item** and save the result in **product** ; otherwise, skip the multiplication. In either case, print the value of **product**.
 - Store the absolute difference of **x** and **y** in **y** , where the absolute difference is (**x - y**) or (**y - x**), whichever is positive. Do not use the **abs** or **fabs** function in your solution.
 - If **x** is 0 , add 1 to **zero_count**. If **x** is negative, add **x** to **minus_sum**. If **x** is greater than 0 , add **x** to **plus_sum**.

Write/paste your code here ▼		

5. Consider the C arithmetic expression $2 * ((i \% 5) * (4 + (j - 3) / (k + 2)))$ where i , j and k are integer variables. If these variables are assigned the values 8, 15 and 4, respectively, then the given determine the value of the expression. (**Note:** The interpretation of the remainder operation (%) is unclear when one of the operands is negative. Most versions of C assign the sign of the first operand to the remainder. The % operation is undefined when second operand is zero.)

Expression evaluation ▼

6. Consider the following C expressions;

- (a) Suppose that i is an integer variable whose value is 7, and f is a floating-point variable whose value is 8.5. The expression $(i + f) \% 4$ is valid or invalid.
- (b) Suppose that i is an integer variable whose value is 7, and f is a floating-point variable whose value is 8.5. The expression $((int)(i + f)) \% 4$ is valid or invalid.

Answer

Cause

(a)

(b)

7. ASCII code for the character ? is 63. Characters are represented by integer codes, C permits conversion of type **char** to type **int** and vice versa. So find the output for the given code snippet;

```
int q_code = (int) '?';
printf("%d %c %d\n", q_code, '?', '?');
```

Output

8. The following expressions contain different operands and operators assuming $x=3.0$, $y=4.0$, and $z=2.0$ are type **double**, $flag=0$ is type **int**. Write each expressions value.

- (a) $!flag$
- (b) $x + y / z <= 3.5$
- (c) $!flag || (y + z >= x - z)$
- (d) $!(flag || (y + z >= x - z))$

Answer

(a)

(b)

(c)

(d)

9. What value is assigned to the type **int** variable **ans** in this statement if the value of **p** is 100 and **q** is 50?

```
ans = (p > 95) + (q < 95);
```

Output

10. Evaluate each of the following expressions if *a* is 6 , *b* is 9 , *c* is 14 , and *flag* is 1 . Which parts of these expressions are not evaluated due to short-circuit evaluation?

- (a) `c == a + b || !flag`
(b) `a != 7 && flag || c >= 6`
(c) `!(b <= 12) && a % 2 == 0`
(d) `!(a > 5 || c < a + b)`

Answer

- (a)
(b)
(c)
(d)

11. Suppose that *i* is an integer variable, *x* is a floating-point variable, *d* is a double-precision variable and *c* is a character-type variable. Find the output generated by these statements that make use of the operator **sizeof**.

```
printf("integer:%ld bytes\n", sizeof i);
printf("integer:%ld bytes\n", sizeof(i));
printf("float:%ld bytes\n", sizeof x);
printf("float:%ld bytes\n", sizeof(x));
printf("double:%ld bytes\n", sizeof d);
printf("double:%ld bytes\n", sizeof(d));
printf("character:%ld bytes\n", sizeof c);
printf("character:%ld bytes\n", sizeof(c));
/* Same way can be used for other data types to
   find the size */
```

Answer

12. Another way to generate the same information as like previous question is to use a cast rather than a variable within each printf statement. Find the output generated by these statements that make use of the operator **sizeof**.

```
printf("integer:%ld bytes\n", sizeof(int));
printf("float:%ld bytes\n", sizeof(float));
printf("double:%ld bytes\n", sizeof(double));
printf("character:%ld bytes\n", sizeof(char));
/* Same way can be used for other data types to
   find the size */
```

Answer

13. C supports several assignment operators. The most commonly used assignment operator is `=`. Assignment expressions that make use of this operator are written in the form **identifier = expression**, where **identifier** generally represents a variable, and **expression** represents a constant, a variable or a more complex expression. Determine the expression values assume that *i* is an integer-type variable, and that the ASCII character set applies.

```
i=('x'-'o')/3;
i=('y'-'o')/3;
i=2*j/2; (say j is an integer and j is 5)
i=2*(j/2);
i=3.0;
i=-3.5;
```

Answer

NOTE: Multiple assignments of the form
identifier 1 = identifier 2 = ... = expression
are permissible in C. In such situations, the assignments are carried out from **right to left**.

14. C also contains other form assignment operators: +=, -=, *=, /=, %= etc., called short hand operators. Suppose that i and j are integer variables whose values are 5 and 7, and f and g are floating-point variables whose values are 5.5 and -3.25. Determine the value of the expressions

```
i += 5;  
f -= g;  
j *= ( i - 3 );  
f /= 3;  
i %= ( j - 2 )
```

Answer

15. Suppose that x, y and z are integer variables which have been assigned the values 2, 3 and 4, respectively. Determine the value of the given expression;

```
x*=-2*(y+z)/3;
```

Answer

16. The assignment statement that contains a conditional expression on the right-hand side. Determine the value of flag if i=-5 and i=-6 respectively.

```
flag = ( i < 0 ) ? 0 : 100
```

Answer

17. In the following assignment statement, a, b and c are assumed to be integer variables. If a, b and c have the values 1, 2 and 3, respectively, then determine the value of the expression that includes operators of different precedence groups.

```
c += ( a > 0 && a <= 10 ) ? ++a : a / b ;
```

Answer

18. Illustrate the purpose of the following code snippet over the inputs a,b and c respectively.

```
int m1,m2,a,b,c;  
printf("Enter the values of a,b,c:");  
scanf("%d%d%d",&a,&b,&c);  
m1=(a>b)?a:b;  
m2=(m1>c)?m1:c;  
printf("%d\n",m2);
```

Answer

19. A C program contains the following declarations and initial assignments:

```
int i= 8;  
int j = 5;  
float x = 0.005;  
float y = -0.01;  
char c = 'c' , d = 'd' ;
```

Determine the value of each of the following expressions. Use the values initially assigned to the variables for each expression.

- (a) (3 * i - 2 * j) % (2 * d - c)
- (b) (x > y) && (i > 0) && (j < 5)
- (c) 2 * x + (y == 0)
- (d) (2 * x + y) == 0
- (e) 5 * (i + j) > ' c '
- (f) i++

Answer

20. Suppose a is an unsigned integer variable (say represented in 16 bits format) whose value is 0x6db7. In the following the expression, we will shift all bits of a six places to the right and assign the resulting bit pattern to the unsigned integer variable b. Find the resulting value of b. Also write the lost bits because of shifting.

b = a >> 6 ;

Answer

21. Determine the value of each of the following expressions, assume that a is an unsigned integer variable whose initial value is 0x6db7.

- (a) a &= 0x7f
- (b) a ^= 0x7f
- (c) a |= 0x7f
- (d) a = a & 0x3f06
- (e) a = a | 0x3f06 << 8

Answer

22. Determine the output of the following code snippet.

```
int main(){
    int m1,a,b,c;
    printf("Enter the values of a,b,c:");
    scanf("%d%d%d",&a,&b,&c);
    m1=a>b?a>c?a:c:b;
    printf("m1=%d\n",m1);
    return 0;
}
```

Answer

- (1) a=10 b=20 30 m1=
- (2) a=30 b=10 c=20 m1=
- (3) a=20 b=30 c=10 m1=

23. Evaluate the expressions;

Assume A, B, num, xy, f, t, p, q, r are int type variables;

- (1) A=10+(num=2)*3;
- (2) B +=(xy *=3); [here xy=10]
- (3) x +=(f=(t*=20)); [here x=20, t=10]
- (4) p=q=r=100;

Answer

- (1) A =
- (2) B =
- (3) x =
- (4) p= q= r=

24. State the output of the following code snippet;

```
int a,b,s;  
s=scanf("%d%d%d",&a,&b,&a);  
printf("%d\n",s+printf("OSW CSE="));
```

Answer

25. State the output of the following code snippet;

```
int i=-1,j=-1,k=0,l=2,m;  
m=++i || k++ && ++j || l++;  
printf("%d %d %d %d %d\n", i,j,k,l,m);
```

Answer

26. State the output of the following code snippet;

```
int i=10,j=6;  
printf("%d\n", i+++j++);
```

Answer

27. State the output of the following code snippet;

```
int i=3>4, j=4>3;  
int k=(i=j);  
int l=(k==j);  
printf("%d %d %d %d",i,j,k,l);
```

Answer

28. State the output of the following code snippet;

```
int x=400;  
printf("%d %d\n",x=40,x>=50);
```

Answer

29. verify the output/ error of the following code snippet;

```
int i=2,j=0;  
int k=i&& j=1;  
printf("%d\n",k);
```

Answer

30. Find the output of the following code snippet;

```
int i=2,j=2;  
int k=i^j&i;  
printf("%d\n",k);
```

Answer

31. Find the output of the following code snippet;

```
int i=3,j=2;  
int k=i << 1 > 5;  
printf("%d\n",k);
```

Answer