

Started on	Friday, November 18, 2022, 8:11 PM
State	Finished
Completed on	Friday, November 18, 2022, 8:36 PM
Time taken	25 mins 2 secs
Grade	100.00 out of 100.00

Information

Don't Forget to Submit

After you've completed, you must use the "**Submit All and Finish**" button to actually submit.

Standard Library Headers

Assume every question has all necessary headers included. Don't choose "**Doesn't Compile**" as an answer only because code fragments don't include necessary standard library headers.

Programming Questions

Pre-check button allows you to compile, link, and execute your code. If the grader seems to freeze or hang with the **Pre-check** button showing an spinning icon, it is most possibly caused by an infinite loop. The simplest solution that will save you time is to kill the process running the bad submission. You can do this by quitting the question and then reattempting it.

Click on **Reset** [at top-right of code window] to clear the code window. Use **CTRL-c** if you want to copy your code to the clipboard before clicking the **Reset** button.

Question **1**

Correct

2.00 points out of 2.00

Write the **exact** text printed to standard output by the following code fragment:

```
int a = 0, b = 1, c = a + b;
printf("%d", (a == b, a != b, a != c == b));
```

Write NC if the code fragment doesn't compile.

Answer:

1

✓

The correct answer is: 1

Question **2**

Correct

4.00 points out of 4.00

Given definitions:

```
char ch = 'a';
short sh = 10;
int i = 10;
```

write the **exact** type that the following expression will evaluate to.

```
ch + sh * i - 10UL
```

Write NC if the compiler finds the above expression to be illegal.

Answer:

unsigned long

✓

The correct answer is: unsigned long

Question **3**

Correct

2.00 points out of 2.00

Write the values printed to standard output by the following code fragment:

```
int a = 5, b = 10, c = 20;
int x = (a <= 20) ? b : (c = 30);
int y = a <= 20 ? b == 10 : c == 20;
printf("%d %d", x, y);
```

10 ✓

1 ✓

Question **4**

Correct

6.00 points out of 6.00

Write the values printed to standard output by the following code fragment:

```
int a = 6; int b = 5; int c = 4;
int d = 3; int e = 2; int f = 1;
a -= b += -c-- +d++ + - -e++ / ++f;
printf("%i %i %i %i %i %i", a, b, c, d, e, f);
```

1 ✓

5 ✓

3 ✓

4 ✓

3 ✓

2 ✓

Question **5**

Correct

10.00 points out
of 10.00

Use the following statement to fill in the subsequent blanks:

```
scanf("Enter your height and weight: %f %f", &height, &weight);
```

(a) There are tokens in the above statement. [4 points]The tokens in the statement `scanf("Enter your height and weight: %f %f", &height, &weight);` are:

1. `scanf`
2. `(`
3. `"Please enter your height and weight: %f %f"`
4. `,`
5. `&`
6. `height`
7. `,`
8. `&`
9. `weight`
10. `)`
11. `;`

One possible correct answer is: 11

(b) There are tokens that are C identifiers in the above statement. [2 points]The tokens in the statement `scanf("Enter your height and weight: %f %f", &height, &weight);` that are C identifiers:

1. `scanf`
2. `height`
3. `weight`

One possible correct answer is: 3

(c) There are tokens that are C punctuators in the above statement. [2 points]The tokens in the statement `scanf("Enter your height and weight: %f %f", &height, &weight);` that are C punctuators:

1. `,`
2. `,`
3. `;`

One possible correct answer is: 5

(d) There are tokens that are C operators in the above statement. [2 points]The tokens that are C operators in the statement `scanf("Enter your height and weight: %f %f", &height, &weight);` are:

1. `(`
2. `)`
3. `&`
4. `&`

One possible correct answer is: 2

Your answer is correct.

Question **6**

Correct

4.00 points out of 4.00

Write the values printed to standard output by the following code fragment:

```
int i, counter;
for (i = counter = 0; i < 50; i += 2, ++counter)
    i += 2;
printf("%d %d", i, counter);
```

52 ✓

13 ✓

Question **7**

Correct

2.00 points out of 2.00

Write the *exact* text printed to standard output by the following code fragment:

```
int a = 3, b = 8, c = 4, d = 1, e = 5;
printf("%d", a > b && b < 10 || ++a == 4);
```

Write NC if the code fragment doesn't compile.

Answer:

1 ✓

The correct answer is: 1

Question **8**

Correct

4.00 points out of 4.00

Write the *exact* text printed to the standard output stream by the following code fragment:

```
int i, sum;
for (i = sum = 0; i < 10; i++)
    if (2 == i % 3) sum += i;
    else continue;
printf("%d", sum);
```

Write NC if the code fragment doesn't compile.

Answer:

15 ✓

The correct answer is: 15

Question **9**

Correct

4.00 points out of 4.00

Write the values printed to standard output by the following code fragment:

```
int w = 1, x = 2, y = 3;
int z = ++w == x ? w = ++x : w == y++;
printf("%d %d %d %d", w, x, y, z);
```

3 ✓

3 ✓

3 ✓

3 ✓

Question **10**

Correct

6.00 points out of 6.00

Indicate whether each of the following identifiers is legal or illegal.

default	Legal	✓
90balloons	Not legal	✓
for#	Not legal	✓
327_	Not legal	✓
_327	Legal	✓
declare	Legal	✓
include	Legal	✓
\$again	Not legal	✓
identifier	Legal	✓
define	Legal	✓
CONST	Legal	✓
Grid.x	Not legal	✓

Your answer is correct.

The correct answer is: **default** → Legal, **90balloons** → Not legal, **for#** → Not legal, **327_** → Not legal, **_327** → Legal, **declare** → Legal, **include** → Legal, **\$again** → Not legal, **identifier** → Legal, **define** → Legal, **CONST** → Legal, **Grid.x** → Not legal

Question **11**

Correct

6.00 points out of 6.00

Write the *exact* values printed to standard output by the following statement:

`printf("%d %d %d", 77, 077, 0x77);`

77

✓

63

✓

119

✓

Question **12**

Correct

4.00 points out of 4.00

Write the *exact* text printed to standard output by the following code fragment:

```
int i, j;
for (i = j = 0; i >= 10; i++, j += 2) ;
    printf("%d%d", i, j);
for (i = 5, j = i-1; i > 0, j > 0; --i, j = i-1)
    printf("%d", i);
```

Write NC if the code fragment doesn't compile.

Answer:

005432

✓

The correct answer is: 005432

Question **13**

Correct

4.00 points out of 4.00

Define a function **max** that returns the greater of two **int** parameters. Use the following use cases to guide your definition:

```
// use cases for function max
printf("%d", max(1, 2)); // prints 2 to standard output
printf("%d", max(-1, -2)); // prints -1 to standard output
```

Note: You must write the definition of function **max** without including any C standard library headers.

```
2 int max(int a, int b)
3 {
4     if(a >= b)
5     {
6         return a;
7     }
8     else
9     {
10        return b;
11    }
12 }
13
```

Correct

Evaluation details:

Evaluation:

-Summary of tests

```
>+-----+
>| 1 test run/ 1 test passed |
>+-----+
```

```
int max(int x, int y) {
    return (x > y) ? x : y;
}
```

Question **14**

Correct

10.00 points out of 10.00

The following [rather confusing] function finds the median [that is, the middle value] of three numbers. Rewrite the function so that it has just one **return** statement.

```
double median(double x, double y, double z) {
    if (x <= y)
    if (y <= z) return y;
    else if (x <= z) return z;
    else return x;
    if (z <= y) return y;
    if (x <= z) return x;
    return z;
}
```

Note: Your definition of function **median** must not include any C standard library headers.

```
2 double median(double x, double y, double z)
3 {
4     double result;
5     if ((x <= y && x >=z)|| (x <= z && x >= y)) result = x;
6     else if ((x >= y && y >=z)|| (x <= y && y <= z)) result = y;
7     else if ((z >= x && z <= y)|| (z<=x && z >= y))result = z;
8
9     return result;
10
11 }
```

Correct

Evaluation details:

Evaluation:

-Summary of tests

>+-----+
>| 1 test run/ 1 test passed |
>+-----+

One [hard] way:

```
double median(double x, double y, double z) {
    double middle;
    if (x <= y) {
        if (y <= z) {
            middle = y;
        } else if (x <= z) {
            middle = z;
        } else {
            middle = x;
        }
    } else {
        if (z <= y) {
            middle = y;
        } else if (x <= z) {
            middle = x;
        } else {
            middle = z;
        }
    }
    return middle;
}
```

Terser way:

```
double median(double x, double y, double z) {
    double middle;
    if (x <= y) {
        middle = (y <= z) ? y : (x<=z) ? z : x;
    } else {
        middle = (z <= y) ? y : (x<=z) ? x : z;
    }
    return middle;
}
```

More terser way:

```
double median(double x, double y, double z) {
    return (x <= y) ? (y<=z ? y : (x<=z) ? z : x) : (z<=y ? y : (x<=z) ? x : z);
}
```

Question **15**

Correct

4.00 points out of 4.00

Replace the following **while** statement with a **for** statement:

```
// assume all variables are of type int
while (++i <= j) {
    sum += i;
}
```

Note: Any use of keyword **do** or keyword **while** will prevent your code from compiling.

```
5  for (i = 0; i <= j ;i++)sum += i;
```

Correct

Evaluation details:

Evaluation:

-Summary of tests

```
>+-----+
>|  1  test run/ 1  test passed |
>+-----+
```

Easy answer:

```
for (; ++i <= j; ) {
    sum += i;
}
```

Another Answer:

```
for (++i; i <= j; ++i) {
    sum += i;
}
```


Question **16**

Correct

4.00 points out of 4.00

Replace the following **for** statement with a **while** statement:

```
// assume i, j, and sum are int variables
for (i = 1; i < j+1; j++, i = j+2) {
    sum += i;
}
```

Note: Any use of keyword **for** or keyword **do** will prevent your code from compiling.

```
7  i = 1;
8  while(i<=j)
9  {
10
11
12    sum += i;
13    ++j;
14    i = j +2;
15
16 }
```

Correct

Evaluation details:

Evaluation:

-Summary of tests

>+-----+
>| 1 test run/ 1 test passed |
>+-----+

```
i = 1;
while ( i < j+1 ) {
    sum += i;
    ++j;
    i = j+2;
}
```

Question **17**

Correct

10.00 points out of 10.00

Suppose a program requires the user to provide only one of the ten digits [that is, digit 0 through digit 9] as valid input. Implement such a program that reports the count of incorrect inputs provided by the client. For example, if the client provides as input the sequence of characters a b 9 [with each character followed by a newline], the program will print 2 [and nothing else] to standard output.

Note: Include all C standard headers necessary to compile and link your code.

```
2  int IsDigit(int ch) {
3      return ('0' <= ch && ch <= '9') ? 1 : 0;
4  }
5  #include <stdio.h>
6  int main(void) {
7      char ch;
8      int non_digit = 0;
9      do {
10         scanf("%c%c", &ch);
11     } while (IsDigit(ch) == 0 && ++non_digit);
12     printf("%d", non_digit);
13     return 0;
14 }
15
```

Correct

Evaluation details:

Evaluation:

-Summary of tests

>+-----+
>| 3 tests run/ 3 tests passed |
>+-----+

```
int IsDigit(int ch) {
    return ('0' <= ch && ch <= '9') ? 1 : 0;
}

#include <stdio.h>

int main(void) {
    char ch;
    int non_digit = 0;
    do {
        scanf("%c%c", &ch);
    } while (IsDigit(ch) == 0 && ++non_digit);
    printf("%d", non_digit);

    return 0;
}
```

Question **18**

Correct

9.00 points out of 9.00

Define a function **is_alpha** that returns a value 1 if the function's parameter is a Latin character; otherwise the function returns 0. A Latin character belongs to either the set of characters [A, Z] or the set of characters [a, z]. Use the following use cases to guide your definition:

```
// use cases for function is_alpha
printf("%c", is_alpha('m')); // prints 1 to standard output
printf("%c", is_alpha('2')); // prints 0 to standard output
printf("%c", is_alpha('M')); // prints 1 to standard output
```

Note: You must write the definition of function **is_alpha** without including any C standard library headers.

```
2 int is_alpha(int ch) {
3     return (('A' <= ch && ch <= 'Z') || ('a' <= ch && ch <= 'z')) ? 1 : 0;
4 }
5
```

Correct

Evaluation details:

Evaluation:

-Summary of tests

>+-----+
>| 1 test run/ 1 test passed |
>+-----+

```
int is_alpha(int ch) {
    return (('A' <= ch && ch <= 'Z') || ('a' <= ch && ch <= 'z')) ? 1 : 0;
}
```

The function's parameter must have type **int** because it must handle the possibility of receiving **EOF**.

Question **19**

Correct

5.00 points out of 5.00

Replace the following iteration statement with a *single statement* that is neither an iteration statement nor a selection statement.

```
while (a < b) {  
    ++a;  
}
```

Note: Any use of keywords **if**, **switch**, **while**, **for**, or **do** in your code will prevent that code from compiling.

```
3  a<b?a=b:a;
```

Correct

Evaluation details:

Evaluation:

-Summary of tests

>+-----+
>| 1 test run/ 1 test passed |
>+-----+

```
a = (a < b) ? b : a;
```