Continental Mathematics League

2018 - 2019

Computer Science Contest

Grades 6-8

The contest consists of three "meets." Each meet has six questions for 30 minutes.

Note: All parts of a problem must be answered correctly for a student to receive credit for the problem. There is no partial credit.

Computer Science Grades 6-8

Meet 1

1. The programming language Python includes an *interactive* command processor. You type a command at the >>> *prompt*, and the interpreter responds. For example:

```
>>> 2**3
8
>>> sum(x**2 for x in (1, 2, 3))
14
```

Fill in the blank:

```
>>> sum(x**2 for x in (1, 2, 3, 4))
```

____

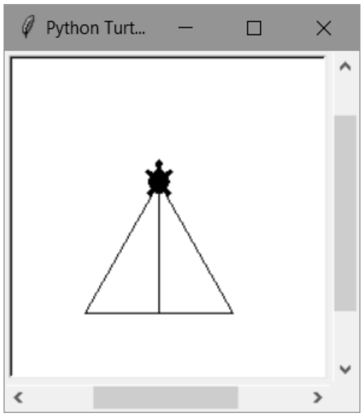
2. In Java (and many other programming languages), == means "is equal to." `a%b` gives the remainder when `a` is divided by `b`. For example, `17%5` gives 2. So `a` is evenly divisible by `b` if `a%b` is 0. What do the following Java statements compute?

```
sum = 0;
for (int n = 1; n <= 1000; n++)
  if (n%2 + n%5 == 0)
    sum += n;  // Add n to sum
```

(A) The sum of all numbers from 1 to 1000 that are divisible by 10
(B) The sum of all numbers from 1 to 1000 that are NOT divisible by 10
(C) The sum of all ODD numbers from 1 to 1000 that are divisible by 5
(D) The sum of all ODD numbers from 1 to 1000 that are NOT divisible by 5
(E) The sum of all numbers from 1 to 1000 that are either EVEN or divisible by 5

Answer: __________

3. Gabi used Python's Turtle Graphics *module* to produce this picture:



Gabi shared a piece of her code that does the actual drawing with a friend. Gabi said that the turtle starts at the lower left corner of the triangle, facing right. Unfortunately, Gabi spilled milk over one of the commands:

```
pendown()      # Prepare to draw
forward(100)   # Move forward 100 units
left(120)      # Turn left 120 degrees
forward(100)
left(120)
forward(100)
left(120)
penup()
forward(50)
left(90)

forward(86)
penup()
```

What is the command that Gabi spilled milk over?

(A) `forward(100)`
(B) `left(90)`
(C) `forward(50)`
(D) `pendown()`

Answer: ______________

4. In Python, the function `str(n)` converts an integer `n` into a string of its digits. For example, `str(123)` makes `'123'`. The function `int(s)` converts a string of digits `s` into an integer. For example, `int('123')` makes `123`. The operator `+`, when applied to two strings, *concatenates* them (puts them together one after the other). For example, `'1' + '2'` makes `'12'`. What does `int(str(3) + str(4)) + 1` make?

   Answer: __________

5. Let's call a number "increasing" if each of its digits, going left to right, is not less than the previous one. For example, 24457. Similarly, let's call a number "decreasing" if each of its digits is not greater than the previous one. For example, 96440. Some numbers are both increasing and decreasing, for example, 3333. If a number is <u>neither</u> increasing nor decreasing, let's call it "bouncy." There are 165 increasing numbers and 219 decreasing numbers among all three-digit numbers (numbers from 100 to 999). How many three-digit bouncy numbers are there?

   Answer: ______________

6. In the Fibonacci number sequence, $F_1, F_2, ..., F_n, ...$, $F_1 = 1, F_2 = 1$, and every subsequent number is the sum of the two previous numbers. So the Fibonacci sequence goes like this: 1, 1, 2, 3, 5, 8, 13, 21, ... (The sequence is named after the Italian mathematician Leonardo of Pisa, known as Fibonacci, who introduced and studied the sequence about 800 years ago.) The Fibonacci sequence comes up again and again in computer science and mathematics.

   Interestingly, the sum of the first *n* Fibonacci numbers is one less than another Fibonacci number. For example, 1+1+2+3+5 = 12 = 13−1 and 1+1+2+3+5+8 = 20 = 21−1. Given that $F_{19} = 4181$ and $F_{20} = 6765$, what is the sum of the first 20 Fibonacci numbers? Hint: Calculate F_{21} and F_{22} first.

   Answer: __________

Computer Science Grades 6-8

Meet 2

1. In the programming language Python, a list has the *methods* `pop` and `append`: `lst.pop(0)` removes the first element of the list `lst` and returns its value; `lst.append(x)` adds `x` at the end of `lst`. For example, if `lst = [1, 2, 3]`, then the statement `x = lst.pop(0)` sets `x` equal to 1 and `lst` becomes `[2, 3]`. If `lst` is `[2, 3]`, then `lst.append(4)` changes `lst` to `[2, 3, 4]`.

   What are the numbers in `lst` after the following Python code runs?

```
lst = [1, 2, 3, 4]
count = 100
while (count > 0):
    x = lst.pop(0)
    lst.append(x)
    count -= 1  # subtract 1 from count
```

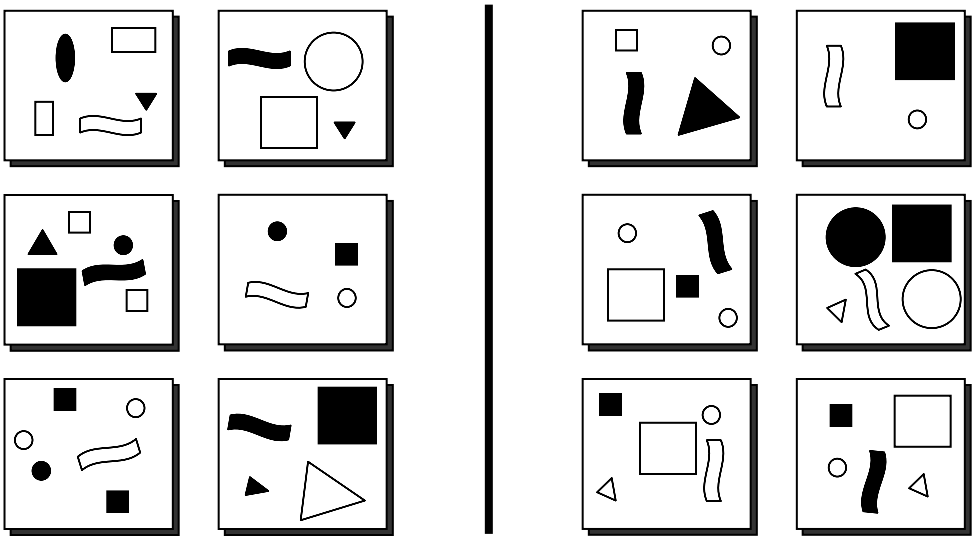
   (A) `[1, 2, 3, 4]`
   (B) `[2, 3, 4, 1]`
   (C) `[3, 4, 2, 1]`
   (D) `[4, 3, 2, 1]`

   Answer: __________

2. A number is *palindromic* if it remains the same when its digits are reversed. For example, 15951. There are 90 three-digit palindromic numbers. How many seven-digit palindromic numbers are there?

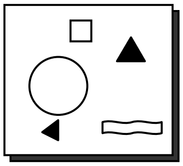
   Answer: __________

3. Mikhail Bongard, a Russian computer scientist, invented a type of puzzle that a computer with AI (short for Artificial Intelligence) was supposed to solve. Here is one puzzle of this type:

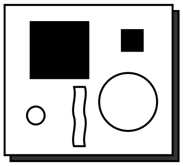


The six pictures to the left of the dividing line are in some way different from the six pictures to the right. Here are five more pictures:

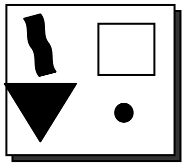
1



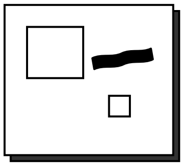
2



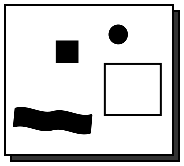
3



4



5



Three of them belong with the left-side pictures. Which ones?

Answer: _________, _________, and _________

4. Ellie wants to confirm her understanding of the difference between `x++` and `++x` in Java. She wrote the following Java code to see how they work:

```
x = 5;
System.out.print(x++);
System.out.println(x);

x = 5;
System.out.print(++x);
System.out.println(x);
```

The output was

```
56
66
```

What output should Ellie expect from the following code?

```
x = 5;
System.out.print(x++);
System.out.println(++x);
```

Answer: __________

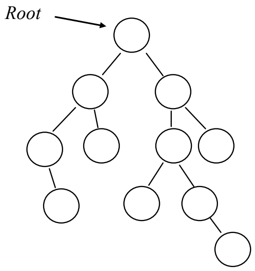
5. You have two glasses, a three-ounce glass and a five-ounce glass. You are allowed to perform three operations:

1. Fill a glass with water.
2. Empty a glass into the sink.
3. Pour as much water as possible from one glass into the other (until the first glass is empty or the second glass is full).

You need to get exactly four ounces in the five-ounce glass, with the three-ounce glass empty. What is the smallest possible number of operations needed?

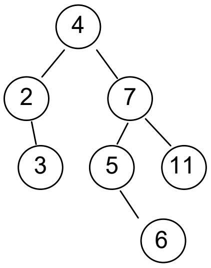
Answer: __________

6. In computer science, *trees* grow upside down, with the *root* above the *leaves*. In a *binary tree*, each *node* is linked to at most two other nodes immediately below it. For example:



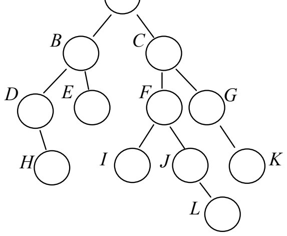
From each node of a binary tree may grow one or two smaller trees, called the *left* subtree and *right* subtree, respectively.

A *binary search tree* holds values in its nodes in such a way that the value in any node is greater than any value in that node's left subtree and less than any value in that node's right subtree. For example:



Duplicate values are not allowed in a binary search tree.

In the binary search tree below, the nodes are labeled with letters, but the values held in the nodes are not shown:



Which letters correspond to the nodes that hold the smallest and the largest values in this tree?

Answer: Smallest __________ Largest __________

Computer Science Grades 6-8

Meet 3

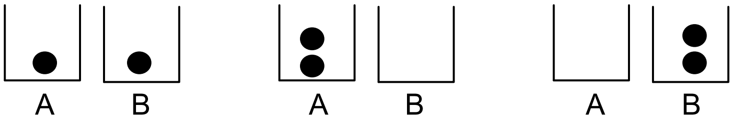
1. The *octal* (base-8) number system uses eight digits, 0 - 7, to represent numbers. The *binary* (base-2) number system uses only 2 digits, 0 and 1. The following table shows a few numbers in octal and binary.

Regular (decimal)	**Octal**	**Binary**
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
...	...	...
20	24	10100
21	25	10101
...	...	...
500	764	111110100

How is octal 765 written in binary?

Answer: ______________________

2. There are three ways to distribute two identical marbles into two boxes:



(a) How many ways are there to distribute three identical marbles into four different boxes? (Some of the boxes will remain empty.)

(b) Now a trick question: How many ways are there to distribute seven identical marbles into four different boxes in such a way that each box holds at least one marble?

Answer: (a) _________ (b) _________

3. Recall that the n-th power of a is the product of n a's: $a^n = \underbrace{a * a * ... * a}_{n}$. (We use $*$ for multiplication, like in computer code.) For example, $a^4 = a * a * a * a$. Raising a number to a power is called *exponentiation*. In a naive exponentiation *algorithm* (method) it takes $n-1$ multiplications to compute a^n. For example, it takes three multiplications to compute a^4:

```
a2 = a * a
a3 = a2 * a
a4 = a3 * a
```

But in an optimized algorithm it takes only two multiplications:

```
a2 = a * a
a4 = a2 * a2
```

How many multiplications are required to compute x^8 using the optimized algorithm?

(A) 7
(B) 6
(C) 4
(D) 3

Answer: ______________

4. The programming language Python has a function `divmod(n, d)`, which returns the quotient and the remainder when `n` is divided by `d`. For example, if you write

```
q, r = divmod(17, 2)
```

`q` gets the value 8 and `r` gets the value 1. If you write

```
q, r = divmod(18, 2)
```

then `q` is 9 and `r` is 0.

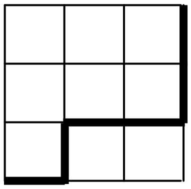
Now consider this *recursive* function (that is, a function that calls itself):

```
def count_something(n):
    q, r = divmod(n, 2)
    if r is 1:
        return 0
    else:
        return count_something(q) + 1
```

`count_something(12)` returns 2. What does `count_something(48)` return?

Answer: ___________

5.



The path in the above picture starts at the lower left corner, proceeds along the grid lines up or to the right, and ends at the upper right corner. There are 20 such paths in the 3 by 3 grid. The area under the path shown above is 2 (squares). What is the sum of the areas below all 20 paths? Hint: Rotate the grid 180º; the path is still there but the area above becomes the area below...

Answer: __________

6. Alice has six cards with the numbers from 1 to 6 written on them. The cards are shuffled and arranged face down. Bob challenges Alice to arrange the cards in order without looking at them: she can only point to any two cards and Bob will tell her which one is smaller. Alice says: Easy. She takes any card and puts it aside. She then takes another card, Bob tells her how it compares with the first, and Alice arranges the two in order. She then takes any other card, excluding the ones she put aside, compares it with the first, and, if necessary, with the second already arranged cards, and inserts the card in its place in order. That's three comparisons so far in the worst case. Alice continues the same way with the remaining cards. Bob says: That way you need 1 + 2 + 3 + 4 + 5 = 15 comparisons in the worst case. Can you do better?

Bob says: Try the following *algorithm* (method). Split the cards into two piles of three and arrange each pile in order, as you did before, with the smallest card on top. Now compare the top two cards and put the smaller one aside. Compare the top two cards in the reduced piles and put the smaller card behind the one already "sorted." Continue this way until one of the piles is empty; then just add the cards in the remaining pile at the end of the "sorted" cards. How many comparisons will you need in the worst case using this algorithm?

Answer: ___________

Computer Science Grades 6-8

Answers and Solutions

Meet 1

1. 30
2. A
3. D
4. 35
5. 525
6. 17710

Solutions:

1. The statement computes the sum of the squares of the numbers in parentheses. Adding 4 to the numbers adds 16 to the sum: 14 + 16 = 30.

2. `n%x >= 0` for non-negative `n`, so `n%2 + n%5 == 0` only if `n%2 == 0` and `n%5 == 0`, that is, `n` is evenly divisible by 2 and by 5.

3. `pendown()` is missing before drawing the altitude of the triangle.

4. `str(3) + str(4)` makes `'3' + '4' = '34'`. `int('34')` gives 34. 34 + 1 = 35.

5. There are 900 three-digit numbers (from 100 to 999). Out of that number subtract the number of increasing and decreasing numbers 900 - 165 - 219 = 516. But there are 9 numbers that are both increasing and decreasing (111, 222, …, 999). We subtracted their count twice. We need to add them back in: 516 + 9 = 525. This computation is a simple case of the *inclusion-exclusion* principle.

6. You can verify that $F_1 + F_2 + ... + F_n = F_{n+2} - 1$. Proof using the method of *mathematical induction*:

   1 + 1 = 3 – 1, so this is true for *n* = 2. If $F_1 + F_2 + ... + F_n = F_{n+2} - 1$, then $F_1 + F_2 + ... + F_n + F_{n+1} = F_{n+2} - 1 + F_{n+1} = F_{n+3} - 1$, so if this is true for *n*, it is also true for *n*+1. By mathematical induction, the statement is true for any *n* >= 2.

   Here $F_{19} = 4181$, $F_{20} = 6765$, $F_{21} = 4181 + 6765 = 10946$; $F_{22} = 6765{+}10946 = 17711$; $F_{22} - 1 = 17710$.

Meet 2

1. A
2. 9000
3. 1, 4, and 5
4. 57
5. 7
6. Smallest: D, Largest: K

Solutions:

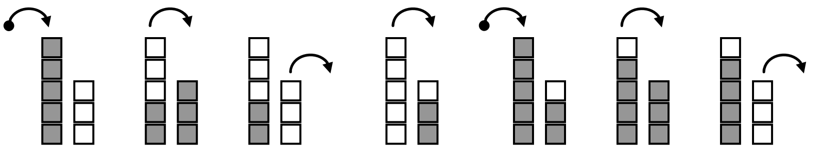
1. Every four iterations restore the list to its original state. 100 is a multiple of four.

2. Only the first four digits matter — the remaining digits are uniquely defined by the palindrome property. So the number of palindromic seven-digit numbers is the same as the number of all four-digit numbers, 1000 through 9999.

3. In the left-side pictures the thin squiggles are nearly horizontal; in the right-side pictures they are nearly vertical.

4. `x++` first uses the value of x, then adds one to x. `++x` first adds one to the value of `x`, then uses the increased value of x. The first print statement displays 5, as in the given example, while `x` becomes 6. In the second print statement, `x` becomes 7, then that new value is displayed.

5.



6 To reach the smallest node follow the links to the left child, as far down as possible. To reach the largest node, follow the links to the right child, as far down as possible.

Meet 3

1. 111110101
2. 20 and 20
3. D (or 3)
4. 4
5. 90
6. 11

Solutions:

1. Each octal digit can be represented by a three-digit binary number. You just need to put together binary equivalents for 7, 6, and 5: 111, 110 and 101.

2. 4 ways when all three marbles are in one box, 4*3 = 12 ways when two marbles are in one box and the third one in a different box, and four ways when all three marbles are in different boxes. In general, there are $\binom{n+m-1}{m}=\binom{n+m-1}{n-1}$ ways to distribute *m* marbles into *n* boxes. The number $\binom{a}{b}=\frac{a!}{b!(a-b)!}$ is called the *binomial coefficient* or "Choose *n*, *k*." It is the number of ways to choose *k* objects out of *n* different objects. Proof of the marbles and boxes formula: Put $n+m-1$ dots in a row. Draw $n-1$ vertical bars through some of the dots. The remaining dots are "marbles" and the partitions between the bars are "boxes."

3.

```
    a2 = a * a
    a4 = a2 * a2
    a8 = a4 * a4
```

4. `count_something(24)` returns `count_something(12)` + 1 = 3. `count_something(48)` returns `count_something(24)` + 1 = 4. In general `count_something(n)` returns the number of times 2 divides `n`, that is, the exponent of the highest power of 2 that divides `n`. For instance, $48 = 3 \cdot 2^4$.

5. If we rotate the grid, the area above becomes the area below, and vice-versa. So all the paths come in pairs, and the sum of the areas in each pair is 9. There are 20 lines — 10 pairs. The sum of all the areas is 10 * 9 = 90.

6. In the worst case: 3 comparisons to sort each pile of 3 cards, plus 5 comparisons to merge the piles into one. 3 + 3 + 5 = 11. The first algorithm is called *Insertion Sort*. The second algorithm, if you apply it *recursively* to piles of any size > 2, is called *Mergesort*.