© 2019 Ruixuan Tu @ AITC

ACSL Test 1

65 MINUTES, 7 QUESTIONS

Turn to your answer sheet to answer the questions in this test.

DIRECTIONS

In this test, you will be able to demonstrate your ability to apply your computer knowledge about a variety of topics. You will answer several questions by writing steps or code. Answer each of the questions as completely as possible.

You may answer the first 5 questions by writing steps, and you may answer the last 2 questions by writing code.

CODE

`ACSL1-2019`

Test begins on the next page.

Do not open the booklet until you are told to do so.

Unauthorized copying or reuse of any part of this page is illegal.

CONTINUE

© 2019 Ruixuan Tu @ AITC

1

Evaluate the following expression and express the answer in octal:

$$2_{10} * 416_8 + A23_{16} - 10110_2 + 31_8 / 101_2$$

2

What is printed when the program is run?

```
A   DC      2
B   DC      1
C   DC      1
D   DC      1
TOP LOAD    B
    MULT    A
    STORE   B
    LOAD    C
    ADD     B
    STORE   C
    LOAD    D
    ADD     =1
    STORE   D
    LOAD    C
    SUB     =100
    BL      TOP
    PRINT   D
    END
```

3

Simplify the following LISP expression:

```
(CDR (CAR (CDR (REVERSE (CAR (CDR '(1 (2 ((3 4) 2) 3) 2 (4 (1 2)))))))))
```

4

What is printed when this program is run?

```
a = 16: b = 2: c = 4: d = 0: e = 5
if a < b then a = a + b else a = a - b
if a * d < c * e then a = a + d else c = c + e
if b ^ d = 1 then d = d + 1 else b = b + 1
if int(a / c) = a / c then a = a / c else a = a - c
if (a < e) or (b >= d) then a = e else b = d
if (b >= c) and (d <= e) then c = b - c else d = a - e
print a ^ d + b * e / a - c * (a / e + b ^ b)
end
```

Unauthorized copying or reuse of any part of this page is illegal.

CONTINUE

5

Start with a segment 16 cm in length. At each end draw a segment half as long, perpendicular to the original segment and intersecting at the new segment's midpoint. If this process is continued infinitely, a fractal is created. Counting the original segment as stage 1, what is the total length of all segments after stage 6 is completed?

6

Code for *Wrap Around Code* using C, C++, or Java.

Wrap Around Code

This is yet another in a long list of ACSL code programs. You would think we would have run out of them by now. In this program you will be given a set of letters to encode. The difference here is that different rules are used for different letters and the counting process starts where the last letter ends. Using the numerical value of each letter (A=1, B=2, ⋯ Z= 26) the rules are as follows:

	The number of letters to count is given by:
RULE 1	Multiply its numerical value by 2.
RULE 2	Divide its numerical value by 3. Multiply the integer remainder by 5.
RULE 3	Divide its numerical value by 4. Multiply the integer part of the quotient by –8.
RULE 4	Take the square root of the numerical value. Truncate the decimal part of the answer. Multiply the resulting integer by –12.
RULE 5	Find the sum of the factors of its numerical value. Multiply by 10.

For the first letter of each set, if the computed value is non-negative, starting at A count the computed value to the right. For each additional letter start at the current encoded letter. If the first computed value is negative, starting at A count to the left which means wrapping around to the end of the alphabet. For each additional letter start at the current encoded letter. As an example, the input C, 1, B, 2, F, 3, $ would produce the encoded letters GQI. The C with a numerical value of 3 evaluates to a 6,

Unauthorized copying or reuse of any part of this page is illegal.

CONTINUE

using rule 1. Always starting each new set at A, and counting 6 letters to the right, the C encodes to an G. The B with a numerical value of 2 evaluates to a 10. Counting down the alphabet 10 letters from G encodes the B to a Q. The F with a numerical value of 6 evaluates to a –8, using rule 3. Counting to the left 8 letters from Q encodes the F to an I. The final encoded value is GQI.

Input Format:

There will be 5 input lines. Each line will consist of a series of upper case letters each followed by a rule number. The line will end with a $. You may enter the letters and numbers one at a time. The commas shown are for clarification and do not have to be entered. The $ is not encoded.

Output Format:

For each input line, print the encoded string it produces.

Sample Input:

```
C,1,B,2,F,3,$
A,1,A,1,A,1,$
A,1,H,2,D,3,$
T,5,S,4,$
```

Sample Output:

```
GQI
CEG
CME
EI
```

7

Code for *Mixing Milk* using C, C++, or Java.

Mixing Milk

Farming is competitive business – particularly milk production. Farmer John figures that if he doesn't innovate in his milk production methods, his dairy business could get creamed!

Fortunately, Farmer John has a good idea. His three prize dairy cows Bessie, Elsie, and Mildred each produce milk with a slightly different taste, and he plans to mix these together to get the perfect blend of flavors.

Unauthorized copying or reuse of any part of this page is illegal.

CONTINUE

To mix the three different milks, he takes three buckets containing milk from the three cows. The buckets may have different sizes, and may not be completely full. He then pours bucket 1 into bucket 2, then bucket 2 into bucket 3, then bucket 3 into bucket 1, then bucket 1 into bucket 2, and so on in a cyclic fashion, for a total of 100 pour operations (so the 100th pour would be from bucket 1 into bucket 2). When Farmer John pours from bucket a into bucket b, he pours as much milk as possible until either bucket a becomes empty or bucket b becomes full.

Please tell Farmer John how much milk will be in each bucket after he finishes all 100 pours.

INPUT FORMAT (file mixmilk.in):

The first line of the input file contains two space-separated integers: the capacity c_1 of the first bucket, and the amount of milk m_1 in the first bucket. Both c_1 and m_1 are positive and at most 1 billion, with $c_1 \geq m_1$. The second and third lines are similar, containing capacities and milk amounts for the second and third buckets.

OUTPUT FORMAT (file mixmilk.out):

Please print three lines of output, giving the final amount of milk in each bucket, after 100 pour operations.

SAMPLE INPUT:

```
10 3
11 4
12 5
```

SAMPLE OUTPUT:

```
0
10
2
```

In this example, the milk in each bucket is as follows during the sequence of pours:

```
Initial State: 3  4  5
1. Pour 1->2:  0  7  5
2. Pour 2->3:  0  0  12
3. Pour 3->1:  10 0  2
4. Pour 1->2:  0  10 2
5. Pour 2->3:  0  0  12
(The last three states then repeat in a cycle ...)
```

Unauthorized copying or reuse of any part of this page is illegal.

CONTINUE

© 2019 Ruixuan Tu @ AITC

STOP

If you finish before time is called, you may check your work on this test only.

Do not work on other things.

No Test Material On This Page

Unauthorized copying or reuse of any part of this page is illegal.