ACSL

2013 - 2014 **American Computer Science League** **All-Star**

Short Round Solutions

1. **Boolean Algebra**

$$\overline{(\overline{A}(B+\overline{C})+B\overline{C})}\ \overline{(A\overline{B}+C+ABC)} = \overline{\overline{A}(B+\overline{C})+B\overline{C}} + \overline{A\overline{B}+C+\overline{ABC}}$$

$$= \overline{\overline{A}(B+\overline{C})} + \overline{B\overline{C}} + \overline{A\overline{B}+C} + \overline{ABC} = \overline{\overline{A}} + \overline{B+\overline{C}} + \overline{B} + \overline{\overline{C}} + \overline{A\overline{B}}\,\overline{C} + \overline{A} + \overline{B} + \overline{C}$$

$$= A + \overline{B}\,\overline{\overline{C}} + \overline{B} + C + (\overline{A}+\overline{\overline{B}})\overline{C} + \overline{A} + \overline{B} + \overline{C} = A + \overline{A} + \overline{B} + \overline{B} + C + \overline{C} + \overline{B}C + (\overline{A}+B)\overline{C}$$

$$= 1 + 1 + \overline{B}(C+1) + (\overline{A}+B)\overline{C} = 1$$

1. (D) 1

2. Digital Electronics

The digital circuit is represented by the following Boolean expression: ~((A◊~AB◊(B+C))((B+C) □CD□~D))

A	B	C	D	~D	~AB	B+C	◊	CD	□	( )( )	~
0	0	0	0	1	1	0	1	0	1	1	0
0	0	0	1	0	1	0	1	0	0	0	1
0	0	1	0	1	1	1	1	0	1	1	0
0	0	1	1	0	1	1	1	1	1	1	0
0	1	0	0	1	1	1	1	0	1	1	0
0	1	0	1	0	1	1	1	0	1	1	0
0	1	1	0	1	1	1	1	0	1	1	0
0	1	1	1	0	1	1	1	1	1	1	0
1	0	0	0	1	1	0	1	0	1	1	0
1	0	0	1	0	1	0	1	0	0	0	1
1	0	1	0	1	1	1	1	0	1	1	0
1	0	1	1	0	1	1	1	1	1	1	0
1	1	0	0	1	0	1	1	0	1	1	0
1	1	0	1	0	0	1	1	0	1	1	0
1	1	1	0	1	0	1	1	0	1	1	0
1	1	1	1	0	0	1	1	1	1	1	0

2. (B) 2

3. Prefix-Infix-Postfix

- + % 9 2 * & 9 ↑ + / # 9 7 4 & 4 2 / / * / * 9 # 7 5 % 7 2 8 & + 9 7 & 4
= - + % 9 2 *(& 9) ↑ + / # 9 7 4 (& 4) 2 / / * / * 9 # 7 5 % 7 2 8 & + 9 7 (& 4)
= - + (% 9 2)* 3 ↑ + / (# 9 7) 4 2 2 / / * / * 9 (# 7 5)(% 7 2) 8 & (+ 9 7) 2
= - + 4 * 3 ↑ + (/ 8 4) 2 2 / / * / (* 9 6)3 8 (& 16) 2
= - + 4 * 3 ↑ (+ 2 2) 2 / / * (/ 54 3) 8 4 2
= - + 4 * 3 (↑ 4 2)/ / (* 18 8)4 2 = - + 4 (* 3 16) /(/ 144 4)2
= -(+ 4 48)(/ 36 2) = - 52 18 = 34

3. (C) 34

4. Computer Number Systems

$11_2 = 3$
$11_8 = 9$
$11011_2 = 27$
$51_{16} = 81$

In base 10 the sequence is:
3 9 27 81 ?
3^1 3^2 3^3 3^4 3^5
So $3^5 = 243 = 363_8$

4. (D) 363

5.Bit String Flicking Let X = abcde

LHS = (LCIRC – 2 (01011 OR abcde)) AND (RSHIFT – 2 (10111 AND abcde))
= (LCIRC – 2 a1c11) AND (RSHIFT – 2 a0cde)
= c11a1 AND 00a0c = 00a0c

RHS = (RCIRC – 2 (LSHIFT – 1 abcde))
= (RCIRC – 2 bcde0) = e0bcd

So 00a0c = e0bcd $\Rightarrow$ a = b, c = d = 0, d = 0, e = 0 $\therefore$ 11000, 00000

5. (C) 11000, 00000

6. What Does This Program Do?

This program counts the number of decimals from 0.01, 0.02, 0.03, …, 0.99 that can be written as a reduced fraction with 100 as the denominator. There are 40 of them. 0.01, 0.03, 0.07, 0.011, …0.97, 0.99.

6. (E) None of the above

7. Recursive Functions

$f(13,1) = f(12,4) + 3 = 88 + 3 = 91$
$f(12,4) = f(11,7) + 3 = 85 + 3 = 88$
$f(11,7) = f(10,10) + 3 = 82 + 3 = 85$
$f(10,10) = 10 + f(14,8) = 10 + 72 = 82$
$f(14,8) = f(13,11) + 3 = 69 + 3 = 72$
$f(13,11) = f(12,14) + 3 = 66 + 3 = 69$
$f(12,14) = 2*12 + 3*14 = 24 + 42 = 66$ Now substitute backwards.

7. (B) 91

8. Data Structures

Depth	Sum at that depth	Internal Length
1	1 x 2 = 2	2
2	2 x 4 = 8	10
3	3 x 8 = 24	34
4	4 x 16 = 64	98
5	5 x 32 = 160	258
6	6 x 64 = 384	642

8. (B) 642

9. Graph Theory

The cycles from A are ADCA, ABCA.
Edge EB adds cycles ADEBCA and AEBCA.
Edge EA adds cycles ABCEA, ADCEA, ADEA, AEA.
Edge EC adds cycles ADECA, AECA.
Edge ED adds cycle AEDCA
Edge BA adds cycle ABA, ADCBA

9. (D) ED

---

10. LISP

(CAR(CAR(REVERSE(CDR(CDR '(1(2 (3 4))(5 (6 7) 8 (9 (1 4))))))))
=(CAR(CAR(REVERSE(CDR '((2 (3 4))(5 (6 7) 8 (9 (1 4)))))))
=(CAR(CAR(REVERSE '((5 (6 7) 8 (9 (1 4)))))))
=(CAR(CAR '((5 (6 7) 8 (9 (1 4))))))
=(CAR '(5 (6 7) 8 (9 (1 4))))
= 5

10. (E) None of the above

---

11. FSA and Regular Expressions

The FSA translates to the following regular expression:

11*0((10*(010*1 OR 110*1)01*)
OR (11*0(10*11 OR 10*0)01*))01*(111*1 OR 10*00)

Only choices a, c, d and f are accepted.

11. (D) a, c, d, f

---

12. Assembly Language

The high level program equivalent to the program is as follows:

```
READ N
WHILE N <> 0 DO
   TEMP1 = 2*X*X
   TEMP2 = 4*X
   ANS = TEMP1 – TEMP2 – 30
   IF ANS = 0 THEN PRINT N
END
```

This program tests some numbers to see if they are solutions to $2x^2 – 4x – 30 = 0$. The roots are -3 and 5.

12. (C) -3, 5