

Questions

0. Predict the output of this snippet:

```
1 #include <stdio.h>
2 int main() {
3     int main = 56;
4     printf("%d", main);
5     return 0;
6 }
```

- (a) Compiler Error
- (b) Depends on the compiler
- (c) 56 ←
- (d) none of the above

1. Predict the output of this snippet:

```
1 #include <stdio.h>
2 int main() {
3     char ch;
4     if (ch = printf("")) {
5         printf("It matters\n");
6     }
7     else {
8         printf("It doesn't matter\n");
9     }
10    return 0;
11 }
```

- (a) It matters
- (b) It doesn't matter ←
- (c) Runtime error
- (d) Nothing

2. How many times is Hello world printed by the code snippet below?

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 int main() {
```

```
5     fork();
6     fork();
7     printf("Hello world\n");
8 }
```

- (a) 1
- (b) 2
- (c) 4 ←
- (d) 8

3. What is the output of this C code?

```
1  #include <stdio.h>
2  int main() {
3      int x = 1, y = 0, z = 5;
4      int a = x && y || ++z;
5      printf("%d", z++);
6  }
```

- (a) 1
- (b) 5
- (c) 6 ←
- (d) 7

4. What is the output of this C code?

```
1  #include <stdio.h>
2  int main() {
3      int y = 2;
4      int z = y +(y = 10);
5      printf("%d\n", z);
6  }
```

- (a) 2
- (b) 4
- (c) 20 ←
- (d) Compile time error

5. What is the output of this C code?

```

1  #include <stdio.h>
2  #define max(a) a
3  int main() {
4      int x = 1;
5      switch (x)
6      {
7          case max(2):
8              printf("yes\n");
9          case max(1):
10             printf("no\n");
11             break;
12     }
13 }

```

- (a) yes
- (b) no ←
- (c) Runtime error
- (d) Compile time error

6. What is the output of this C code?

```

1  #include <stdio.h>
2  int main() {
3      int x = 35;
4      printf("%d %d %d", x == 35, x = 50, x > 40);
5      return 0;
6  }

```

- (a) 1 50 1
- (b) 0 50 0 ←
- (c) Runtime error
- (d) Compile time error

7. You have two numbers, A and B.

Using bitwise operations and loops, write code to determine how many bits must be flipped in order to turn A in to B.

e.g.

A: 101001

B: 100101

Answer: 2 bits

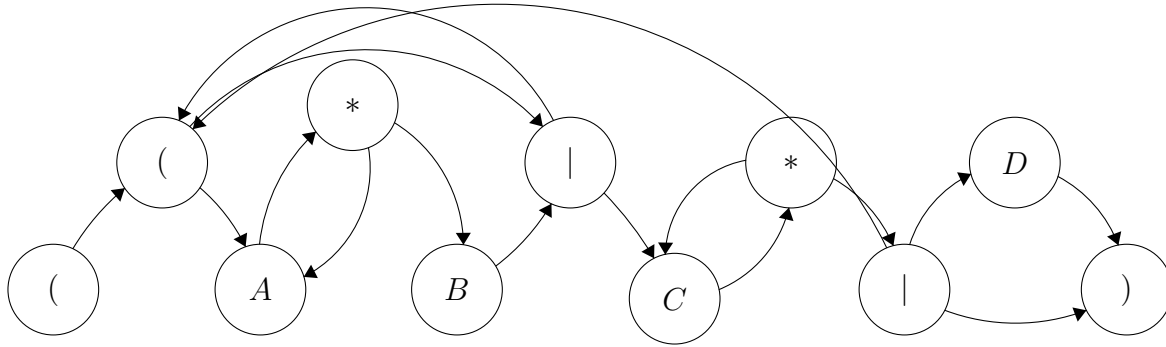

```

5     int    lenMax          = 0;
6     // Max of subarray end at Index
7     int    sumMaxEndHere   = 0;
8     int    startIdxMaxEndHere = 0;
9     int    lenMaxEndHere   = 0;
10
11     int i;
12     for (i=0; i < len; i++) {
13         // Find sumMaxEnd at this Index
14         if (sumMaxEndHere > 0) {
15             sumMaxEndHere   = sumMaxEndHere + *(array+i);
16             //startIdxMaxEndHere = startIdxMaxEndHere;
17             lenMaxEndHere   = lenMaxEndHere + 1;
18         }
19         else {
20             sumMaxEndHere   = *(array+i);
21             startIdxMaxEndHere = i;
22             lenMaxEndHere   = 1;
23         }
24         if (sumMaxEndHere > sumMax) {
25             sumMax          = sumMaxEndHere;
26             startIdxMax     = startIdxMaxEndHere;
27             lenMax          = lenMaxEndHere;
28         }
29     }
30     if (pArrayOut) {
31         *pArrayOut = array + startIdxMax;
32     }
33     if (pLenOut) {
34         *pLenOut = lenMax;
35     }
36     return sumMax;
37 }

```

10. RE pattern matching.

Draw an NFA (nondeterministic finite state automaton) that recognizes the same language that the regular expression $((A^*B|C)^*|D)$



NFA for $((A^*B|C)^*|D)$