Hard-Level Pseudocode MCQs – Selection (Decision Control)

Only one choice is guaranteed correct.

All variables are 32-bit signed integers unless stated otherwise.

Pay attention to side effects inside conditions and compound expressions.

1.

```
x ← 7
if (x++ == 8) ++x else x--
print x
```

A. 6

B. 7

C. 8

D. 9

2.

```
a ← 5
b ← 3
if (a & b) then
    a ← a + 1
else if (a | b)
    a ← a - 1
else
    a ← 0
print a
```

A. 4

B. 5

C. 6

D. 7

3.

```
n ← -2147483648        // INT_MIN
if (n < 0 && -n < 0)
    print "A"
else
```

```
    print "B"
```

A. A

B. B

C. Runtime error

D. Undefined

4.

```
x ← 10
if (x = 0)            // single '='
    print x
else
    print x + 1
```

A. 0

B. 1

C. 10

D. 11

5.

```
p ← 0
q ← 1
if (p && (q = 5))
    q++
print q
```

A. 1

B. 2

C. 5

D. 6

6.

```
a ← 2
b ← 3
c ← 4
if (a < b < c)
    print "YES"
else
```

```
    print "NO"
```

A. YES

B. NO

C. Syntax error

D. Garbage

7.

```
mask ← 0x80000000
if (mask & (mask - 1))
    print 1
else
    print 0
```

A. 0

B. 1

C. -1

D. 2147483647

8.

```
k ← 5
select
    case k & 3:
        0: print "A"
        1: print "B"
        2: print "C"
        3: print "D"
end select
```

A. A

B. B

C. C

D. D

9.

```
x ← 3
if (x << 31)
    print "LEFT"
```

```
else
    print "RIGHT"
```

A. LEFT

B. RIGHT

C. Implementation-defined

D. Overflow exception

10.

```
a ← 7
b ← 0
if (a && (b = a / 0))
    print "SAFE"
else
    print "UNSAFE"
```

A. SAFE

B. UNSAFE

C. Divide-by-zero exception

D. Infinite loop

11.

```
s ← "abc"
t ← "abcd"
if (s < t)          // lexicographic
    print len(s)
else
    print len(t)
```

A. 3

B. 4

C. 0

D. 1

12.

```
flag ← true
if (!flag == flag)
    print "EQUAL"
```

```
else
    print "NOT"
```

A. EQUAL

B. NOT

C. Compile-time error

D. Depends on language

13.

```
x ← 1
if (x ^= x)
    print x
else
    print x + 1
```

A. 0

B. 1

C. 2

D. -1

14.

```
a ← 4
b ← 2
if (a & (a - 1))
    b ← b << 1
else
    b ← b >> 1
print b
```

A. 1

B. 2

C. 4

D. 8

15.

```
c ← 0
for i ← 0 to 31 do
    if (1 <<i< 0)
```

```
        c ← c + 1
end for
print c
```

A. 0

B. 1

C. 15

D. 32

16.

```
x ← 100
if (x == (x & -x))
    print "POWER"
else
    print "NOT"
```

A. POWER

B. NOT

C. Implementation-defined

D. Syntax error

17.

```
p ← 3
q ← 4
r ← 5
if (p < q < r == (r > q > p))
    print "SAME"
else
    print "DIFF"
```

A. SAME

B. DIFF

C. Compile-time error

D. Runtime exception

18.

```
a ← 6
b ← 3
```

```
if ((a & b) && (a | b))
    print (a ^ b)
else
    print (a + b)
```

A. 5

B. 6

C. 9

D. 12

19.

```
x ← 8
if (x == (x & (x - 1)))
    print 0
else
    print 1
```

A. 0

B. 1

C. -1

D. 2

20.

```
m ← 0x7FFFFFFF
if (m + 1 > 0)
    print "POS"
else
    print "NEG"
```

A. POS

B. NEG

C. Implementation-defined

D. Runtime trap

Answer Key

1 B

2 A

3 A

4 D

5 A

6 A

7 B

8 D

9 B

10 B

11 A

12 B

13 B

14 A

15 B

16 B

17 A

18 A

19 A

20 B