

## Hard-Level Pseudocode MCQs – Looping Constructs

Only one choice is guaranteed correct.

All variables are 32-bit signed integers unless stated.

Watch for side effects, short-circuit tricks, and overflow wrap-around.

---

1.

```
x ← 0
for i ← 2147483647 downto -2147483648 do
  x ← x + 1
end for
print x
```

- A. 0
  - B.  $2^{31}$
  - C.  $2^{32}$
  - D. 4294967296
- 

2.

```
cnt ← 0
i ← 1
while i > 0 do
  i ← i << 1
  cnt ← cnt + 1
end while
print cnt
```

- A. 31
  - B. 32
  - C. 33
  - D. Infinite loop
- 

3.

```
s ← 0
for k ← 0 to 63 do
  if (1 << k) < 0 then break
end for
```

print k

- A. 30
  - B. 31
  - C. 32
  - D. 63
- 

4.

```
x ← 1
repeat
  x ← -x
until x > 0
print x
```

- A. -1
  - B. 0
  - C. 1
  - D. Never terminates
- 

5.

```
i ← 0
j ← 0
while i < 10
  j ← j + i
  i ← i + (j & 1)
end while
print j
```

- A. 20
  - B. 25
  - C. 30
  - D. 36
- 

6.

```
n ← 1000
while n & (n - 1) ≠ 0 do
  n ← n & (n - 1)
end while
```

print n

- A. 0
  - B. 512
  - C. 768
  - D. 1000
- 

7.

```
sum ← 0
for i ← 0; i ≤ 100; i ← i + (i == 0 ? 1 : i) do
    sum ← sum + i
end for
print sum
```

- A. 5050
  - B. 197
  - C. 1275
  - D. 377
- 

8.

```
a ← 1
b ← 2
while a ≠ b do
    a ← a << 1
    b ← b << 1
end while
print a
```

- A. 0
  - B. 1
  - C.  $2^{31}$
  - D. Loop never ends
- 

9.

```
cnt ← 0
for i ← 0 to 255 do
    for j ← 0 to 255 do
        if (i & j) == 0 then cnt ← cnt + 1
```

```
    end for
end for
print cnt
```

- A. 256
  - B. 6561
  - C. 32768
  - D. 65536
- 

10.

```
x ← 1
for i ← 1 to 31 do
    x ← (x << 1) ^ (x >> 31 & 1)
end for
print x
```

- A. 0x00000001
  - B. 0x7FFFFFFF
  - C. 0x80000000
  - D. 0xFFFFFFFF
- 

11.

```
i ← 0
while true do
    if (i & -i) == i then break
    i ← i + 1
end while
print i
```

- A. 0
  - B. 1
  - C. 2
  - D. 4
- 

12.

```
x ← 0
for i ← 0 to 30 do
    x ← x | (1 << (30 - i))
```

```
    if  $x < 0$  then break
end for
print i
```

- A. 0
  - B. 1
  - C. 30
  - D. 31
- 

13.

```
n ← 12345
rev ← 0
while n ≠ 0 do
    rev ← (rev << 1) | (n & 1)
    n ← n >> 1
end while
print rev
```

- A. 12345
  - B. 11163
  - C. 5349
  - D. 6172
- 

14.

```
c ← 0
for i ← 1 to 100 do
    for j ← 1 to i do
        if  $(i \wedge j) < (i \& j)$  then c ← c + 1
    end for
end for
print c
```

- A. 1617
  - B. 1717
  - C. 1817
  - D. 1917
- 

15.

```
x ← 0
for b ← 0 to 30 do
  x ← x ^ ((1 << b) * (b % 2))
end for
print x
```

- A. 0x2AAAAAAAA
  - B. 0x55555555
  - C. 0xAAAAAAAA
  - D. 0xFFFFFFFF
- 

16.

```
i ← 0
j ← 1
while j > 0 do
  j ← j << 1
  i ← i + 1
end while
print (1 << i) == j
```

- A. true
  - B. false
  - C. Loop never ends
  - D. Depends on sign bit
- 

17.

```
sum ← 0
for i ← 0 to 63 do
  if (i & (i >> 1)) == 0 then sum ← sum + i
end for
print sum
```

- A. 1089
  - B. 1156
  - C. 1225
  - D. 1296
- 

18.

```
x ← 0x5A5A5A5A
cnt ← 0
repeat
    x ← x & (x - 1)
    cnt ← cnt + 1
until x == 0
print cnt
```

- A. 8
  - B. 12
  - C. 16
  - D. 20
- 

19.

```
x ← 1
for i ← 1 to 5 do
    x ← x * 3
    if x ≥ 100 then continue
    x ← x + 1
end for
print x
```

- A. 243
  - B. 244
  - C. 245
  - D. 246
- 

20.

```
i ← 0
j ← 0
while i < 256
    j ← j + __builtin_popcount(i) // counts 1-bits
    i ← i + 1
end while
print j
```

- A. 512
- B. 1024
- C. 2048
- D. 4096

---

### Answer Key

1 C

2 B

3 B

4 D

5 B

6 B

7 D

8 D

9 B

10 D

11 B

12 B

13 B

14 B

15 B

16 B

17 A

18 C

19 A

20 B