

5) for (int i = 1; i <= 3; i++)

for (int j = 1; j <= 2; j++)

printf(" * ");

A) * * *

B) * * * * *

C) * * * * *

~~d) 6 stars in one line~~

6) int i = 0;

while (i < 3)

printf("%d", ++i);

A) 0 1 2

~~B) 1 2 3~~

C) 1 1 1

d) 0 1 2 3

7) int i = 0;

do {

i++;

} while (i < 0);

printf("%d", i);

A) 0

~~B) 1~~

C) no output

d) infinite loop

8) int i;

for (i = 1; i <= 5; i++);

printf("%d", i);

A) 5

~~B) 6~~

C) 1

d) error

1) for (int i=0; i<5; i++)
 printf("%d", i);

~~A) 0 1 2 3 4~~

B) 1 2 3 4 5

C) infinite loop

D) 0 1 2 3 4 5

2) How many times will this loop execute

int i=1;

✓ while (i <= 5)
 i++;

A) 4 times

✓ B) 5 times

C) 6 times

D) infinite loop

3) int i=5;

while (i--)

 printf("%d", i);

A) 5 4 3 2 1

✓ B) 4 3 2 1 0

C) 4 3 2 1

D) infinite loop

4) int i=1;

do {

 printf("%d", i);

} while (i++ < 3);

✓ A) 1 2 3

B) 1 2

C) 1

D) 1 2 3 4

9) `int i = 1;`
`while (i < 5)`
`i += 2;`
`printf("%d", i);`

- A) 1 B) 2 C) 4 ~~D) 8~~

10) `int i;`
`for (i = 1; i <= 5; i += 2)`
`printf("%d", i);`

A) 1 2 3 4 5

~~B) 1 3 5~~

C) 2 4

D) 1 3

11) `int i = 2`

`while (i-- > 0)`
`printf("%d", i);`

A) 2 1 0

B) 1 0

C) 2 1

~~D) 10 1~~

12) `for (int i = 10; i > 0; i -= 3)`
`printf("x ");`

A) 3

B) 4

~~C) 5~~

D) infinite

13) `int i = 0;`
`while (i++ < 3)`

`printf("%d", i);`

A) 1 2 3

B) 0 1 2

☒ C) 1 2 3 4

D) 0 1 2 3

14) for (int i=1; i<=3; i++)
for (int j=1; j<=i; j++)
printf("*");

☒ A) *** **

B) 6 stars

C) 3 stars

D) Error

15) int n=1;
while (n<10)
n = n+n;
printf("%d", n);

A) 8

B) 10

☒ C) 16

D) infinite

16) int i=5;
for (; i>0;)
i--;
printf("%d", i);

A) 5

B) 4

☒ C) 0

D) Error


```
17) int i=1;  
do {  
    printf("%d", i);  
    i += 2;  
} while(i < 6);
```

- A) 135
- B) 13
- C) 13 57
- D) 1

```
18) int i=0;  
while (++i < 3)  
    printf("%d", i);
```

- A) 12
- B) 0123
- C) 123
- D) 23

```
19) int i=10;  
for (; i >= 7; i--)  
    printf("%d", i);
```

- A) 10 9 8 7
- B) 10 9 8
- C) 7 8 9 10
- D) 10 9 8 7 6

```
20) int i=0;  
for (; ; )  
{  
    if (i == 3) break;  
    printf("%d", i);  
    i++;  
}
```

- A) 0 1 2 3
- B) 0 1 2
- C) infinite loop
- D) no output