

1. Jakou hodnotu musí mít proměnná **x**, aby byla podmínka splněna?

```
int x = ...;

if ( x == 28 & 10 )
{
    ...
}
```

→ podmínku nelze splnit pro žádnou hodnotu x

→ 1

→ 8

→ 28

→ podmínka je splněna pro libovolnou hodnotu x

```
/* & je bitový AND ( = binární součin, logické „a zároveň“, tzn. u např.
 * 01100 & 10111 je potřeba, aby jednička byla u obou čísel na stejném
 * místě, tzn. výsledek bude 00100 )
 * nejprve se provede x == 28 → to může být F/T → 1 nebo 0
 * neboli bude to buď 1 & 10 nebo 0 & 10
 * tzn.
 *
 * 0001
 * 1010
 * ----
 * 0000 = 0 → podmínku nelze splnit
 *
 * 0000
 * 1010
 * ----
 * 0000 = 0 → podmínku nelze splnit
 */
```

} tzn. podmínku nelze splnit pro žádnou hodnotu x

2. Jakou hodnotu musí mít proměnná **x**, aby byla podmínka splněna?

```
int x = ...;

if ( x == 13 && 24 )
{
    ...
}
```

→ 8

→ podmínka je splněna pro libovolnou hodnotu x

→ 1

→ 13

→ podmínku nelze splnit pro každou hodnotu x

3. Jakou hodnotu musí mít proměnná **x**, aby byla podmínka splněna?

```
int x = ...;

if ( x == ( 23 || 15 ) )
{
    ...
}
```

→ podmínku nelze splnit pro žádnou hodnotu x

→ buď 23, nebo 15

→ 1

→ **31**

→ podmínka je splněna pro libovolnou hodnotu x

4. Jakou hodnotu musí mít proměnná **x**, aby byla podmínka splněna?

```
int x = ...;  
  
if ( x == 18 && 16 )  
{  
    ...  
}
```

- podmínku nelze splnit pro žádnou hodnotu x
- 1
- 18
- 16
- podmínka je splněna pro libovolnou hodnotu x

5. Jakou hodnotu musí mít proměnná **x**, aby byla podmínka splněna?

```
int x = ...;  
  
if ( x == 7 & 20 )  
{  
    ...  
}
```

- podmínku nelze splnit pro žádnou hodnotu x
- 1
- 7
- 4
- podmínka je splněna pro libovolnou hodnotu x

6. Jakou hodnotu musí mít proměnná **x**, aby byla podmínka splněna?

```
int x = ...;  
  
if ( x == ( 21 & 30 ) )  
{  
    ...  
}
```

- podmínku nelze splnit pro žádnou hodnotu x
- 1
- 21
- 20
- podmínka je splněna pro libovolnou hodnotu x

7. Jakou hodnotu musí mít proměnná **x**, aby byla podmínka splněna?

```
int x = ...;  
  
if ( x == ( 24 & 14 ) )  
{  
    ...  
}
```

- podmínku nelze splnit pro žádnou hodnotu x
- 1
- 24
- 8
- podmínka je splněna pro libovolnou hodnotu x

8. Jakou hodnotu musí mít proměnná **x**, aby byla podmínka splněna?

```
int x = ...;

if ( x == 5 | 12 )
{
    ...
}
```

→ podmínku nelze splnit pro žádnou hodnotu **x**

→ 1

→ 13

→ buď 5, nebo 12

→ podmínka je splněna pro libovolnou hodnotu **x**

*/\* | je bitový OR → stačí, aby byla přítomna alespoň jedna jednička \*/*

9. Co zobrazí následující program?

```
int a = 2, b = 6;

if ( a!=b )
    printf ( "%d", a + 15 );
else
    printf ( "%d", a + 27 );
```

→ Odpověď ( celé desítkové číslo ): 27

*/\* a!=b ← do „a“ se dosadí negace „b“; platí pro každé číslo kromě 0 \*/*

10. Co zobrazí následující program?

```
int a = 2, b = 0;

if ( a!=b )
    printf ( "%d", a + 15 );
else
    printf ( "%d", a + 27 );
```

→ Odpověď ( celé desítkové číslo ): 16

*/\* !(0)=1; 1+15=16 \*/*

11. Co zobrazí následující program?

```
int a = 9, b = 7;

if ( a=b )
    printf ( "%d", a + 12 );
else
    printf ( "%d", a + 25 );
```

→ Odpověď ( celé desítkové číslo ): 19

*/\* do „a“ se přiřadí „b“=7, což je pravda a je to 7+12=19 \*/*

## 12. Co zobrazí následující program?

```
Program:
int x = 73;
if ( 123 > x > 47 )
    printf ( "1" );
else
    printf ( "2" );
```

→ zápis je syntakticky nesprávný

→ [2](#)

→ 1

→ chování programu není definované

→ program spadne

*/\* x = 73; 123 > 73 ... 1; 1 > 47 ... 0 - podmínka nesplněna → 0 \*/*

## 13. Co zobrazí následující program?

```
int a = 27, b = 21;

if ( a != b )
    printf ( "%d", a + 10 );
else
    printf ( "%d", a + 26 );
```

→ Odpověď ( celé desítkové číslo ): [37](#)

## 14. Co zobrazí následující program?

```
int a = 25, b = 29;

if ( a == b )
    printf ( "%d", a + 15 );
else
    printf ( "%d", a + 28 );
```

→ Odpověď ( celé desítkové číslo ): [53](#)

## 15. Co zobrazí následující program?

```
int x = 86;

if ( 129 > x > 14 )
    printf ( "1" );
else
    printf ( "2" );
```

→ 1

→ program spadne

→ [2](#)

→ chování programu není definované

→ zápis je syntakticky nesprávný

16. Co zobrazí následující program?

```
int x = 79;

if ( 134 > x > 21 )
    printf ( "1" );
else
    printf ( "2" );
```

- 1
- program spadne
- chování programu není definované
- [2](#)
- zápis je syntakticky nesprávný

17. Co zobrazí následující program?

```
int a = 21, b = 21;

if ( a == b )
    printf ( "%d", a + 12 );
else
    printf ( "%d", a + 25 );
```

- Odpověď ( celé desítkové číslo ): [33](#)

18. Co zobrazí následující program?

```
int x = 121;

if ( 89 > x < 76 )
    printf ( "1" );
else
    printf ( "2" );
```

- [1](#)
- program spadne
- chování programu není definované
- 2
- zápis je syntakticky nesprávný

*/\* x = 121; 89 > 121 ... 0; 0 < 76 ... 1 - podmínka splněna → 1 \*/*

19. Co zobrazí následující program?

```
int x = 45;

if ( 15 < x < 53 )
    printf ( "1" );
else
    printf ( "2" );
```

- [1](#)
- program spadne
- chování programu není definované
- 2
- zápis je syntakticky nesprávný

*/\* x = 45; 15 < 45 ... 1; 1 < 53 ... 1 - podmínka splněna → 1 \*/*

20. Co zobrazí následující program?

```
int a = 22, b = 28;

if ( a == b )
    printf ( "%d", a + 12 );
else
    printf ( "%d", a + 27 );
```

→ Odpověď ( celé desítkové číslo ): [49](#)

21. Datový typ int je realizován pomocí 32 bitových čísel, pro zobrazení záporných čísel se používá dvojkový doplněk. Kolikrát proběhne následující cyklus?

```
unsigned int i;

for ( i = 3362; i > 0; i-- )
{
    printf ( "x" );
}
```

→ [3362](#)

→ 3361

→ 3363

→  $2^{32}$

→ cyklus je nekonečný

*/\* unsigned → pouze kladná čísla \*/*

22. Datový typ int je realizován pomocí 32 bitových čísel, pro zobrazení záporných čísel se používá dvojkový doplněk. Kolikrát proběhne následující cyklus?

```
int i = 4434;

do
{
    printf ( "x" );
    i --;
} while ( i != 4434 );
```

→  $2^{31}$

→  $2^{32}-1$

→ 4434

→ [2<sup>32</sup>](#)

→ cyklus je nekonečný

*/\* po dosazení 0 se odečte 1, n bude mít hodnotu -1, -2, -3, ... až -2miliardy (binárně je to 10000000000...000) a po dalším odečtení se to číslo zmenší na 011111111...11111, tedy stane se z něj +2miliardy a bude se postupně zmenšovat až do těch 4465. Proměnná „i“ tedy vystřídá všech  $2^{32}$  hodnot \*/*

23. Datový typ `int` je realizován pomocí 32 bitových čísel, pro zobrazení záporných čísel se používá dvojkový doplněk. Kolikrát proběhne následující cyklus?

```
int i = 2115;

do
{
    printf ( "x" );
    i --;
} while ( i != 2115 );
```

- $2^{31}$
- $2^{32}-1$
- 2115
- [2<sup>32</sup>](#)
- cyklus je nekonečný

24. Jaká bude hodnota proměnné `x`?

```
int x;
x = 80;
x *= 1.5;
```

- Odpověď ( desetinné číslo ) : [120](#)

25. Datový typ `int` je realizován pomocí 32 bitových čísel, pro zobrazení záporných čísel se používá dvojkový doplněk. Kolikrát proběhne následující cyklus?

```
int i = -27;
unsigned int y = 593;

if ( x > y )
    printf ( "1" );
else
    printf ( "2" );
```

- program spadne
- zápis je syntakticky nesprávný
- chování programu není definované
- [1](#)
- 2

26. Datový typ `int` je realizován pomocí 32 bitových čísel, pro zobrazení záporných čísel se používá dvojkový doplněk. Kolikrát proběhne následující cyklus?

```
int i;

for ( i = 4457; i >= 0; i-- )
{
    printf ( "x" );
}
```

- [4458](#)
- 4456
- 4457
- $2^{32}$
- cyklus je nekonečný

27. Datový typ `int` je realizován pomocí 32 bitových čísel, pro zobrazení záporných čísel se používá dvojkový doplněk. Kolikrát proběhne následující cyklus?

```
unsigned int i = 4254;

do
{
    printf ( "x" );
    i --;
} while ( i != 4254 );
```

- 4254
- $2^{32}-1$
- $2^{31}$
- [2<sup>32</sup>](#)
- cyklus je nekonečný

28. Datový typ `int` je realizován pomocí 32 bitových čísel, pro zobrazení záporných čísel se používá dvojkový doplněk. Kolikrát proběhne následující cyklus?

```
unsigned int i;

for ( i = 3525; i >= 0; i -- )
{
    printf ( "x" );
}
```

- 3524
- 3525
- 3526
- $2^{32}$
- [cyklus je nekonečný](#)

29. Jaká bude hodnota proměné `y`?

```
int x;
double y;

x = y = 16.5 + 0.0;
```

- Odpověď ( desetinné číslo ) : [16.5](#)

30. Jaká bude hodnota proměnné `y`?

```
int x;
double y;

x = y = 14.625 + 0.0;
```

- Odpověď ( desetinné číslo ) : [14.625](#)



31. Jaká bude hodnota proměnné y?

```
int x;  
double y;  
  
x = y = 15.75 + 0.0;
```

→ Odpověď ( desetinné číslo ) : [15.75](#)

32. Jaká bude hodnota proměnné y?

```
int x;  
double y;  
  
y = x = 13.8125 + 0.0;
```

→ Odpověď ( desetinné číslo ) : [13](#)

*/\* přiřazuje se „int“ do „double“ tzn. proběhne zaokrouhlení \*/*

33. Jaká bude hodnota proměnné y?

```
int x;  
double y;  
  
y = x = 14.6875 + 0.0;
```

→ Odpověď ( desetinné číslo ) : [14](#)

34. Jaká bude hodnota proměnné y?

```
int x;  
double y;  
  
x = y = 18.1875 + 0.0;
```

→ Odpověď ( desetinné číslo ) : [18.1875](#)

35. Jaká bude hodnota proměnné x?

```
int x;  
x = 90;  
x *= 1.5;
```

→ Odpověď ( desetinné číslo ) : [135](#)

*/\* 90\*1.5 \*/*