

1. Mit ír ki az alábbi program?

```
std::cout << !(true && !(true || false));
```

- a. 1
- b. 0
- c. semmit, mert le sem fordul
- d. futásidejű hiba

2. Mit ír ki az alábbi program?

```
int a = 5;  
int b = 5;  
int sum = -(-a-b);  
std::cout << sum;
```

- a. 0
- b. 5
- c. 10
- d. -10

3. Mit ír ki az alábbi program?

```
int N = 10;  
int main()  
{  
    static int x = 1;  
    if(std::cout << x << " " << && x++ < N && main()) { }  
}
```

- a. fordítási idejű hiba, mert a main()-ben kihagytuk a return 0; utasítást
- b. végtelen ciklusba kerülünk, mert az x változó újra és újra egyre fog inicializálódni
- c. 1-től – 10-ig a számokat
- d. nem ír ki semmit, lefut a feltétel és mivel a törzs üres ezért a program befejezi a futást

4. Mit ír ki az alábbi program?

```
int x = 10;  
int y = 10;  
if(~(10^10)) std::cout << "hello"  
else std::cout << "world"
```

- a. hello
- b. world
- c. fordítási idejű hiba
- d. hello world

5. Hány byte-on tárol a C++ egy karaktert? (char)

- a. implementáció függő
- b. 4
- c. 1
- d. 8

6. Melyik **nem** preprocesszor direktíva?

- a. #else
- b. #elif
- c. #undef
- d. #while

7. Melyik definíció az alábbiak közül?

- a. class MyClass;
- b. int a[10];
- c. extern int i;
- d. struct MyStruct;

7. Melyik igaz az alábbiak közül?

- a. az automatikus változók a stack-en jönnek létre
- b. az automatikus változók a statikus tárterületen jönnek létre
- c. az automatikus változók a winchesteren jönnek létre
- d. az automatikus változók a heap-en jönnek létre

9. Mi a csillagozott sorban meghívott művelet neve?

```
class Foo { .. };  
Foo f;  
Foo g = f; /*
```

- a. destruktork
- b. default konstruktor
- c. értékadó operátor
- d. copy konstruktor

10. Az alábbi példában a Foo f(5); konstruktor hívása után mennyi lesz az f.b értéke?

```
struct Foo  
{  
    int a, b;  
    Foo(int c) : a(c*2), b(c*3) {}  
};
```

- a. 10
- b. nemdefiniált viselkedés, mert nincs szekvencia pont
- c. 0
- d. 15

11. Melyik típusnak van push\_front tagfüggvénye?

- a. `std::list`
- b. `std::set`
- c. `std::vector`
- d. egyiknek se

12. Melyik értékadás szabályos az alábbi kód után?

```
int i = 10;  
const int j = 15;  
const int* p = &j;
```

- a. `p = *j;`
- b. `*p = i;`
- c. `p *= i;`
- d. `p = &i;`

13. Melyik konténer asszociatív?

- a. `std::deque`
- b. `std::set`
- c. `std::vector`
- d. egyik sem, mind a három szekvenciális

14. Mit jelent a static kulcsszó az alábbi osztálydefinícióban?

```
struct S { static int x; };
```

- a. S-ből nem lehet objektumot létrehozni
- b. az x változót csak S tagfüggvényei érhetik el
- c. `x` osztálysintű adattag
- d. semmit, struct kulcsszóval nem lehet osztályt definiálni

15. Melyik állítás igaz egy **konstans** objektum esetén?

- a. az objektumnak csak a konstans tagfüggvényei hívhatóak meg
- b. az objektumnak csak private adattagja lehet
- c. az objektum csak default konstruktorral hozható létre
- d. egyik sem, mert fordítási idejű hibát kapunk, ha konstans objektumon szeretnénk tagfüggvényt hívni

16. Mitől válik egy osztály absztrakttá?

- a. van virtuális destruktora
- b. van tisztán virtuális tagfüggvénye
- c. a tagfüggvényeinek csak a deklarációja ismert
- d. van bázisosztálya

17. Az X::f() függvényhívás során mit ír ki a program?

```
int i = 1;
namespace X
{
    int i = 2;
    void f()
    {
        int a = i + 1;
        int i = ::i - 1;
        std::cout << a << ", " << i << std::endl;
    }
}
```

- a. 3, 0
- b. semmit, fordítási hiba keletkezik
- c. 3, 2
- d. 2, 1

18. Mit ír ki az alábbi program?

```
std::cout << true + true << std::endl;
```

- a. nem definiált
- b. fordítási idejű hiba
- c. futás idejű hiba, fordításkor nem derül ki, mert a true típusok megegyeznek
- d. 2

18. Mit ír ki az alábbi program?

```
int i = 3;
int j = i++;
std::cout << j << std::endl;
```

- a. 3
- b. 4
- c. nem definiált
- d. memória szemetet