

Przeanalizuj kod A i B, i odpowiedz na poniższe pytania:

1) W którym przykładzie zmienna bricks będzie zachowywać swoją wartość dla kolejnych wywołań funkcji?

a) A b) B c) A+B d) żadnym

2) W którym z tych przykładów będzie ona widoczna przez inne moduły (inne pliki źródłowe)?

a) A b) B c) A+B d) żadnym

3) Który z poniżej podanych przykładów jest thread-safe?

a) A b) B c) A+B d) żaden

A)

```
static unsigned int bricks;
unsigned int lessbrickly()
{
    if (bricks)
        bricks--;
    return bricks;
}
unsigned int morebricksplease()
{
    return bricks++;
}
```

B)

```
unsigned int bricks( int modf )
{
    static unsigned int bricks;
    if (modf>0){
        if (bricks)
            bricks--;
    }
    else {
        bricks++;
    }
    return bricks;
}
#define lessbrickly() bricks(-1)
#define morebricksplease() bricks(1)
```

Przeanalizuj kod C, i odpowiedz na poniższe pytania zakładając, że kod jest skompilowany na 32 bitowy x86.:

- 4) jaka będzie wartość zwrócona przez funkcję blah(401);
a) 42364 b) 0x193 c) 0x7f d) 403
- 5) jaka będzie wartość zwrócona przez funkcję blah((unsigned short) -1);
a) 0x7f7f b) 32788 c) 32781 d) -2
- 6) ile będzie wynosić wartość pola f4 jeśli blah(402);
a) 402 b) 20 c) 21 d) 2

C)

```
typedef struct foo {
    unsigned int f1:15;
    unsigned int f2:3;
    long f3:10;
    unsigned char f4:3;
} fool;

unsigned int blah( unsigned int a )
{
    struct foo sfoo;
    sfoo.f2 = a;
    sfoo.f4 = a;
    sfoo.f1 = a;

    return sfoo.f2 + sfoo.f4 + sfoo.f1;
}
```

- 7) Wymień 3 specyficzne dla Windows CE kontrolki:
a) SIP, ListView, TodayScreen
b) CommandBar, TodayScreen, SIP
c) SIP, TodayScreen, PropertySheet
- 8) Czy w WindowsCE mogą być uruchomione dwie instancje tej samej aplikacji z tej samej lokalizacji?
a) Tak,
b) Nie (druga po określeniu przez system, że to kopia jest zamykana po inicjalizacji),
c) Może
- 9) Czy Windows CE 2002 obsługuje wyjątki C++?
a) Nie,
b) Tak,
c) Tylko structured exceptions.

- 10) Czemu służy enkapsulacja?
- a) Zamknięciu funkcji w metodzie
 - b) Ograniczeniu dostępu do danych metodom innych obiektów
 - c) Łączeniu klas w obiekty
- 11) Czy referencje dostępne są w C?
- a) tak
 - b) nie
- 12) Klasa `std::vector` jest?
- a) obiektem enkapsulującym wektor fizyczny
 - b) kolejką fifo
 - c) klasą `template`, służącą do przechowywania dowolnych danych
- 13) `std::sort` sortuje:
- a) liczby całkowite
 - b) liczby zmiennoprzecinkowe
 - c) obiekty `std::vector`
 - d) obiekty STL
- 14) Program `gdb` w Linuksie służy do:
- a) interpretacji skryptów w BASICu
 - b) debugowania programów
 - c) kalkulacji liczb o nieograniczonej precyzji
 - d) transferu plików poprzez port równoległy
- 15) Licencja GPL pozwala na wykorzystanie kodu na tej licencji w innych produktach:
- a) pod warunkiem, iż kod wynikowy będzie wydany na licencji GPL
 - a) nawet, jeśli produkt jest na innej licencji, nie musi być opensource
- 16) funkcja `strlen()` z parametrem `NULL` zwróci:
- a) spowoduje błąd `0xC0000005`(windows), `SIGSEGV` (Unix/Linux)
 - b) wartość nieokreśloną
 - c) zero
- 17) czy operator **delete** zeruje wskaźnik?
- a) tak
 - b) tylko, jeśli wskazuje na typ podstawowy
 - c) nie
 - d) zależy od implementacji