

A) Trudność: Harder, Numer indeksu .....

1. Uzupełnij kod operacji tak by sumowanie odbywało się po składowych. Warunek: sumowanie następuje do zmiennej po lewej stronie +.

```
class point{int x; int y;};
point & operator+(point & a, point & b) {
    a.x += b.x;
    a.y += b.y;
    return a;
}
```

2. Podać wzór na wartość dziesiętną liczby w formacie IEEE754, opisać oznaczenia (mantysa, wykładnik, co w jakim formacie, etc.)

$$(-1)^z 2^a m$$

m- wartość mantysy z przoduującym 1. a – wartość liczby w formacie BIAS-127, z – wartość bitu znaku

3.  $1011\ 0011_{(BIAS-4)} = 1011\ 0011_{(NBC)} - 4_{(10)} = 179 - 4_{(10)} = 175_{(10)}$

$1\ 10000011\ 010100000000000000000000_{(IEEE754)} = (-1)^1 2^4 (1 + \frac{5}{16}) = -21_{(10)}$

$-70_{(10)} = -64 - 6 = -1(64 + 6) = -1 * 64 (1 + \frac{6}{64}) = 1\ 10000101\ 000110 \dots 0_{(IEEE754)}$

4. Uzupełnij stos TCP/IP

Warstwa aplikacji
Warstwa internetu
Warstwa transportowa
Warstwa łącza danych / fizyczna

5. Rozwinąć skrót w kontekście baz danych:

<b>A</b>	Atomowość
<b>C</b>	Spójność
<b>I</b>	Izolacja
<b>D</b>	trwałość

6. Wybierz słowa kluczowe ANSI SQL99 (wykreślić które nie są)

SELECT	<del>HURT</del>	AGGREGATE	WHERE	'GROUP BY'	<del>FILTER</del>	SUM	COUNT	NOT
AND	UNION	EXCEPT/MINUS	DISTINCT	<del>WHILE</del>	'UNION ALL'		'ORDER BY'	
JOIN	'CROSS JOIN'	<del>'FUZZY JOIN'</del>	OR	AND	<del>EVALUATE</del>	<del>'BORK UP'</del>	'OUTER JOIN'	
FROM	LIKE	IN	INTERSECT	'INTERSECT ALL'	BETWEEN			

7. Przeanalizuj poniższy kod i odpowiedz na pytania:

- a. Czy kod się skompiluje? a)TAK b) NIE  
b. Co wypisze konsola? a)OK b)NOT OK c) Nic bo kod się nie skompiluje  
c. Podaj wartość zmiennych: a) first.length = 7 b) second.length = 7 c) first.text = „aaaabbb”

```
1 struct Wrapper {
2     int length;
3     std::string text;
4 };
5
6 Wrapper operator+(Wrapper & left, Wrapper & right) {
7     return Wrapper{ left.length + right.length, left.text + right.text };
8 }
9
10 int main()
11 {
12     Wrapper a = { 4, "aaaa" };
13     Wrapper b = { 3, "bbb" };
14     Wrapper first = a + b;
15     Wrapper second = operator+(a, b);
16     std::cout
17         << (first.length == second.length && first.text == second.text ? "OK" : "NOT OK")
18         << std::endl;
19     return 0;
20 }
21
```