

Test wiedzy w wersji C++

XVI OIJ, zawody I stopnia, tura testowa
6 października 2021



Należy założyć, że podane fragmenty kodu są fragmentami programu, którego reszta jest poprawnie napisana (w szczególności są poprawnie zadeklarowane biblioteki i zmienne).

1. Która instrukcja służy do sprawdzenia warunku i (jednokrotnego) wykonania instrukcji zależnie od tego czy ten warunek jest spełniony?

- condition
- if
- check
- true
- while

2. W jaki sposób zapisać w zmiennej x długość napisu zapisanego w zmiennej y typu `string`?

- `x == y->size;`
- `x <- length(y);`
- `x = y.size();`
- `x := y->length;`
- `set(x, len(y));`

3. W jaki sposób obliczyć wynik działania a^3 ?

- `a ^ 3`
- `exp(a, 3)`
- `a * a * a`
- `a * 3`

4. Rozważmy następujący fragment programu:

```
vector<int> elems;  
...  
int p = 0;  
for (int i = 1; i < elems.size(); i++)  
    if (elems[i] > elems[p])  
        p = i;
```

Co zawiera zmienna p po wykonaniu powyższego kodu?

- wartość największego elementu z `elems`
- liczbę elementów `elems`
- liczbę elementów `elems` większych niż `elems[0]`
- pozycję pierwszego największego elementu z `elems`

5. Które z poniższych fragmentów obliczają 33 procent liczby zapisanej w zmiennej x typu `double`?

- `x / 3`
- `x * 0.33`
- `x % 33`
- `x / 100 * 33`
- `33 % x`
- `x * 33%`

6. Rozważmy poniższą funkcję:

```
int oblicz(int n) {  
    int wynik = 0;  
    for (int i = 1; i <= n; i++)  
        wynik += n / i;  
    return wynik;  
}
```

Jaki wynik zwróci wywołanie `oblicz(25)`?

7. Ile gwiazdek wypisze poniższy fragment programu?

```
for (int i = 0; i <= 5; i++)  
    for (int j = 0; j <= 5; j++)  
        cout << '*';
```

8. Rozważmy poniższą funkcję:

```
string funkcja(const string& napis) {  
    string wynik;  
    for (int i = 1; i < napis.size(); i += 2)  
        wynik.push_back(napis[i]);  
    return wynik;  
}
```

Jaki będzie wynik wywołania `funkcja("olimpiada")`?



9. Rozważmy poniższą funkcję:

```
int funkcja(int a, int b) {
    int wynik = 0;
    while (b > 0) {
        wynik += a;
        b--;
    }
    return wynik;
}
```

Zakładając, że zmienne a oraz b są całkowite, dodatnie i nie przekraczają 100, który z poniższych kodów jest równoważny wywołaniu funkcja(a , b)?

- a / b
- $a - b$
- $a + b$
- $a * b$

10. Ile równa jest najmniejsza wspólna wielokrotność liczb 1,2,3,...,10?

11. Rozważmy poniższą funkcję:

```
int rad(int n) {
    int wynik = 1;
    for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) {
            wynik *= i;
            while (n % i == 0)
                n /= i;
        }
    }
    if (n > 1)
        wynik *= n;
    return wynik;
}
```

Ile jest liczb naturalnych n mniejszych niż 255, dla których $\text{rad}(n) == 10$?

12. Rozważmy poniższą funkcję:

```
int js(int n) {
    int wartosc = 1;
    for (int i = 1; ; i++) {
        wartosc *= i;
        if (wartosc > n)
            return i - 1;
    }
}
```

Jaka jest największa wartość parametru n , dla której $\text{js}(n) == 4$?

13. Rozważmy poniższą funkcję:

```
int mx(const vector<int>& tab, int x) {
    int wynik = 0;
    for (int y : tab)
        wynik = max(wynik, x ^ y);
    return wynik;
}
```

Niech

```
tab = vector<int>({11, 10, 24, 26}).
```

Jaką najmniejszą wartość może zwrócić wywołanie $\text{mx}(\text{tab}, x)$ dla odpowiednio dobranej nieujemnej całkowitej wartości x ?

14. Rozważmy poniższą funkcję:

```
bool ws(const vector<int>& tab, int p, int x) {
    if (p >= tab.size())
        return false;
    if (tab[p] == x)
        return true;
    return ws(tab, p + 1, x);
}
```

Dla jakich zawartości tab wywołanie $\text{ws}(\text{tab}, 2, 7)$ zwróci true?

- $\text{vector<int>}(\{3, 7, 1, 7, 5, 2\})$
- $\text{vector<int>}(\{4, 9, 4, 1\})$
- $\text{vector<int>}(\{1, 2, 7\})$
- $\text{vector<int>}(\{5, 8, 0, 2, 7\})$
- $\text{vector<int>}(\{2\})$
- $\text{vector<int>}(\{7, 1, 2, 3, 4\})$

15. Celem poniższej funkcji jest zmodyfikować tab , aby odwrócić kolejność jej elementów. Jak można uzupełnić warunek w ..., aby to osiągnąć?

```
void odwracaj(vector<int>& tab) {
    int n = tab.size();
    for (int i = 0; ...; i++)
        swap(tab[i], tab[n - 1 - i]);
}
```

- $i < n$
- $i < n / 2$
- $i <= n / 2$
- $i < n - 1 - i$
- $i <= n$

16. Ile zer ma liczba $20^{50} \cdot 50^{20} \cdot 90^{90}$ na końcu zapisu dziesiętnego?

17. Poprawnym nawiasowaniem nazywamy napis, który może powstać z wyrażenia arytmetycznego przez opuszczenie wszystkiego poza znakami nawiasów. Na przykład napis $()((()))$ jest poprawnym nawiasowaniem, mógł powstać na przykład z wyrażenia $(2 + 2) * (1 + (2 + 3) * 4)$. Celem poniższej funkcji jest zbadać czy przekazany napis jest poprawnym nawiasowaniem.

```
bool czy_poprawne_nawiasowanie(const string& naw) {
    int min_balans = 0;
    int balans = 0;
    for (char x : naw) {
        if (x == '(')
            balans++;
        else if (x == ')')
            balans--;
        else
            return false;
        min_balans = min(min_balans, balans);
    }
    return ...;
}
```

Jak należy uzupełnić ..., aby to osiągnąć?

- balans == 0
 - (min_balans < balans) && (balans >= 0)
 - min_balans == 0
 - (min_balans >= 0) && (balans == 0)
 - min_balans <= balans
18. Jaka jest najmniejsza liczba naturalna n , której zapis w systemie rzymskim ma dokładnie 10 znaków? Odpowiedź podaj w systemie dziesiętkowym.

19. Ile zmiennych typu string przechowujących napisy długości 1000000 znaków można zadeklarować w programie, aby zużycie pamięci (związane z przechowywaniem tych zmiennych) wynosiło około 50 MB?
- około pięćset
 - około pięciu
 - kilka tysięcy
 - około pięćdziesiąt
 - jedną

20. Rozważmy poniższy fragment programu:

```
for (int i = 1; i <= n; i++)
    for (int j = 1; j * j <= n; j++)
        cout << '*';
```

Jaką liczbę gwiazdek wypisaną przez program można osiągnąć, odpowiednio dobierając n ?

- 400
- 512
- 350
- 256
- 150

21. Rozważmy poniższą funkcję:

```
void wykonuj(int n) {
    for (int i = 1; i <= n; i++) {
        int j = i;
        while (j % 2 == 0) {
            j /= 2;
            cout << "*";
        }
    }
}
```

Zakładamy, że operacje arytmetyczne oraz porównania na zmiennych typu int zajmują czas stały. Jaka jest pesymistyczna złożoność obliczeniowa funkcji wykonuj?

- $\Theta(n \log n)$
- $\Theta(n)$
- $\Theta(n^2)$
- $\Theta(\sqrt{n})$
- $\Theta(\log n)$
- $\Theta(n\sqrt{n})$

22. Które z poniższych zbiorów liczb można podzielić na dwa zbiory o równej sumie? Każdy element powinien trafić do dokładnie jednego zbioru.

- {2, 3, 4, 5, 6, 7, 8}
- {1, 2, 3, 5, 8, 13}
- {1, 2, 3, 4, 5, 6, 7}
- {1, 2, 4, 8, 16, 32, 65}
- {2, 3, 5, 7, 11}

23. Która z podanych liczb jest największa?

- FF_{16}
- 200
- 1111111_2
- $A0_{11}$

24. Ile jest całkowitych nieujemnych kwot, których nie można wydać nominami 7 oraz 11?

25. Dla jakich wartości parametru n poniższa funkcja poprawnie kończy swoje działanie (tzn. nie powoduje błędu wykonania)?

```
void wykonuj(int n) {
    if (n == 100)
        return;
    wykonuj(n + 2);
}
```

- 1000
 10
 100
 1

26. Ile tablic rozmiaru 1 000 000 składających się z liczb naturalnych można posortować z użyciem funkcji `std::sort` na przeciętnym komputerze z roku 2020, aby czas wykonania był równy około pół sekundy? Zakładamy, że sortowania wykonywane są jedno po drugim bez współbieżności.

- kilka tysięcy
 nie można wykonać nawet jednego
 kilka
 kilkaset

27. 60 dzieci ustawiło się w kółku i zaczęło odliczać do dwóch (tzn. pierwsze dziecko mówi 1, drugie dziecko mówi 2, trzecie 1, czwarte 2, piąte 1 itd.). Każde dziecko, które mówi 2 natychmiast wypada z kółka, a wyliczanka jest dalej kontynuowana. Po pełnym okrążeniu przeprowadzana jest analogiczna wyliczanka tylko do trzech (tzn. dzieci liczą 1,2,3 i każde dziecko, które mówi 3 wypada). W kolejnych okrążeniach następują analogiczne wyliczanki, ale do 4,5,6,... W pewnym momencie wyliczanka jest do większej liczby niż liczba dzieci i nikt już dalej nie odpadnie. Ile dzieci pozostanie w kółku?

28. Rozważmy następującą funkcję:

```
int oblicz(const string& operacje) {
    int wynik = 0;
    for (char x : operacje) {
        if (x == '+')
            wynik++;
        if (x == '*')
            wynik *= 2;
    }
    return wynik;
}
```

Podaj napis złożony z czterech znaków + oraz czterech znaków *, który należy przekazać jako parametr operacje, aby `oblicz(operacje) == 35`?

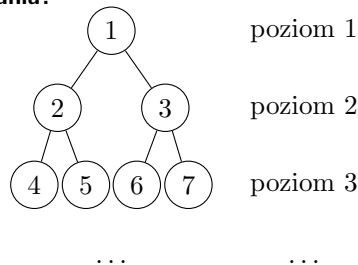
29. Rozważmy poniższą funkcję:

```
int oblicz(int n, int k) {
    int wyniki[n + 1][k + 1];
    for (int i = 0; i <= n; i++)
        for (int j = 0; j <= k; j++)
            wyniki[i][j] = 0;

    for (int i = 0; i <= n; i++) {
        wyniki[i][0] = 1;
        for (int j = 1; j <= min(k, i); j++) {
            wyniki[i][j] += wyniki[i - 1][j];
            wyniki[i][j] += wyniki[i - 1][j - 1];
        }
    }
    return wyniki[n][k];
}
```

Jaki jest wynik wywołania `oblicz(10, 5)`?

30. Na poniższym rysunku wierzchołki są na trzech poziomach, każdy wierzchołek na poziomie innym niż ostatni jest połączony z dwoma na poziomie kolejnym. Wierzchołki są numerowane od 1 kolejno poziomami, a na każdym poziomie od lewej do prawej. Rozważmy podobny rysunek, ale o sześciu poziomach. Ile wynosi największa możliwa do osiągnięcia suma wartości wierzchołków na ciągu kolejno połączonych wierzchołków bez zwracania?



Fragment pełnego drzewa binarnego (do trzech poziomów).