

# Funkcje - odpowiedz na pytanie

Napisz program, który odpowie poprawnie na zadane pytania.

**Pytanie 1.** Które zdanie jest nieprawdziwe:

1. Funkcja to podprogram, który wykonuje pewne zadanie
2. Funkcje pomagają rozwiązywać duże problemy
3. Funkcje to oddzielne aplikacje działające w systemie Windows
4. Funkcje mogą zwracać lub nie zwracać wartości

**Pytanie 2.** Co wypisze poniższy program:

```
#include<iostream>

using namespace std;

void funkcja(int &a, int b)
{
    a = a + b;
    --b;
}

int main()
{
    int a = 3, b = 7;
    funkcja(a, b);
    cout<<a<<" "<<b;

    return 0;
}
```

Odpowiedzi:

1. 10 7
2. 3 7
3. 3 6
4. 10 6

**Pytanie 3.** Funkcja typu inline oznacza, że:

1. Funkcja jest napisana w jednej linii
2. Funkcja działa, gdy nie ma dostępu do Internetu
3. Kod funkcji jest wrzucany w miejsce, gdzie została ona wywołana
4. Kod funkcji jest bardzo krótki

**Pytanie 4.** Które zdanie jest prawdziwe:

1. Funkcja typu **void** zwraca 0
2. Funkcja typu **void** jest zawsze inline
3. W funkcja **void** nigdy nie wystąpi słowo **return**.
4. Funkcja **void** nie zwraca wartości

**Pytanie 5.** Lista argumentów funkcji to:

1. Zmienne globalne programu
2. Niezbędne dane, które przekazane są do funkcji poprzez nawias.
3. Lista instrukcji funkcji
4. Lista nazw funkcji, które wystąpiły w programie.

**Pytanie 6.** Które zdanie jest nieprawdziwe:

1. Funkcja może zwracać wiele wartości
2. Funkcja zwraca wartość poprzez słowo kluczowe **return**
3. Jeśli funkcja zwraca liczbę całkowitą, to poprawny jest zapis: **int x = funkcja();**
4. Jeśli funkcja zwraca wartość, to poprawny jest zapis: `cout<<funkcja();`

**Pytanie 7. Który program jest prawidłowy:**

Program pierwszy:

```
#include<iostream>

using namespace std;

void funkcja(int a, int b)

{

    a = a + b;

}

int main()

{

    int x = funkcja(3, 4);

    return 0;

}
```

Program drugi:

```
#include<iostream>

using namespace std;

void funkcja(int a, int b)

{

    a = a + b;

}

int main()

{

    int x = funkcja(3);

    return 0;

}
```

Program trzeci:

```
#include<iostream>

using namespace std;

void funkcja(int a, int b)

{

    a = a + b;

}

int main()

{

    int x = funkcja(3);

    return 0;

}
```

Program czwarty:

```
#include<iostream>

using namespace std;

int funkcja(int a, int b)

{
```

```
return a + b;
}
int main()
{
int x = funkcja(3,5);
return 0;
}
```

**Pytanie 8\***. Przeciążanie nazw funkcji polega na:

1. Tworzeniu funkcji o tych samych nazwach, ale z różną liczbą argumentów
2. Tworzeniu bardzo długich nazw funkcji
3. Wywoływaniu funkcji wiele razy
4. Nadaniu funkcji bardzo dużej liczby argumentów.

## Input

Jedna liczba z przedziału [1..8] oznaczająca numer pytania

## Output

Odpowiedź na zadane pytanie [1..4]. Jest tylko jedna prawidłowa odpowiedź.

## Example

**Input:**

9

**Output:**

1