

Laboratorium 1.

Zadanie 1.

Napisz, skompiluj i uruchom program, który wyświetla komunikat: "Mam na imie".

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Mam na imie Ania" << endl;
    return 0;
}
```

Zadanie 2.

Dany jest program:

```
// Autor: Jan Kowalski
// Wersja: 1.0

#include <iostream>
using namespace std;
int main() {
    int liczba;
    cout << "Wpisz liczbe: ";
    cin >> liczba;
    cout << "Wpisales: "
        << liczba
        << ' '
        << "Gratuluje, to twój drugi program w C++!"
        << endl;
    return 0;
}
```

Wpisz ten program do pliku, skompiluj go i uruchom. Czy można zmniejszyć liczbę operatorów << tak, aby zachować ten sam układ graficzny wyświetlanego wyniku?

Zadanie 3.

Znajdź wszystkie błędy w poniższym programie:

```
main(x) {
    cin >> "Ile błędów jest w tym programie?" >> liczba_bledow
    cout << W programie znaleziono << liczba_bledow<< endl;
}
```

Zadanie 4.

Znajdź błąd w poniższym programie. Czy ten błąd jest wychwycony przez kompilator? Jak poprawić ten błąd?

```
#include <iostream>
using namespace std;

int main() {
    double a,b;
    int robocza;

    cout << "Wpisz dwie liczby rzeczywiste: ";
    cin >> a >> b;
    // Zamien liczby miejscami
    robocza = a;
    a=b;
    b=robocza;
    cout << a << " " << b << endl;
    return 0;
}
```

Zadanie 5.

Dany jest program:

```
#include <iostream>
using namespace std;
double srednia(double x, double y);

int main()
{
    double a,b;
    cout << "Wpisz dwie liczby: ";
    cin >> a >> b;
    cout << "Srednia z " << a << " i " << b << " to "
    << srednia(a,b) << endl;
    return 0;
}

double srednia (double x, double y)
{
    return (x+y)/2.;
}
```

- Z ilu funkcji zbudowany jest program?
- Wpisz, skompiluj i uruchom program.
- Zakomentuj wiersz zawierający prototyp funkcji. Skompiluj program i przyjrzyj się komunikatom kompilatora. Czy program skompilował się?
- Usuń prototyp i przenieś definicję funkcji powyżej `main`. Czy teraz program skompiluje się?
- Zmodyfikuj program tak, aby prosił o trzy liczby i obliczał ich średnią.

Zadanie 6.

Co będzie wyświetlone na ekranie (i dlaczego) w wyniku działania poniższych instrukcji?

```
cout << 5./10 << endl;  
cout << 5/10 << endl;  
cout << 1/2*10 << endl;  
cout << 1./2*10 << endl;  
cout << 1/2.*10 << endl;  
cout << 5%10 << endl;  
cout << -5%10 << endl;
```

Zadanie 7.

Napisz funkcję, która będzie obliczać BMI (indeks masy ciała):

```
double BMI(int wzrost, int waga);
```

Funkcja ta pobiera wzrost osoby w cm i jej wagę w kg (pełnych). BMI jest obliczany jako *waga w kg podzielona przez kwadrat wzrostu w metrach*. Napisz program, który prosi użytkownika o podanie wagi i wzrostu, wywołuje funkcję BMI i wyświetla obliczony indeks.

Zadanie 8

Napisz funkcję, która obliczy liczbę prostokątnych płytek potrzebnych do ułożenia w pokoju prostokątnym.

Parametry:

Długość i szerokość pokoju, długość i szerokość płytki. Wszystkie wymiary w cm.

W funkcji `main` należy wczytać parametry i wyświetlić wynik.

Zadania domowe

Zadanie 9.

Napisz funkcję, która zamienia wiek psa na odpowiadający mu wiek człowieka:

```
int LudzkiWiekPsa(int lata);
```

Założmy, że pierwszy rok życia psa odpowiada 13 latom życia człowieka. Następnie każde trzy lata w życiu psa odpowiadają 16 latom w życiu człowieka. Funkcja powinna zwrócić wiek psa wyrażony w latach życia człowieka, obcięty do najbliższej liczby całkowitej.

Napisz program, który prosi użytkownika o podanie wieku psa w latach, wywoła funkcję `LudzkiWiekPsa` i wyświetli wynik.

Zadanie 10

Wypełnij puste miejsca w poniższych funkcjach, następnie uzupełnij plik o funkcję main tak, by dało się przetestować ten program.

```
bool cyfra(char d)
// zwraca true, jeśli d jest cyfrą,
// w przeciwnym wypadku zwraca false
{
return .....
}
```

```
bool litera(char c)
// zwraca true, jeśli c jest literą,
// w przeciwnym wypadku zwraca false
{
return .....
}
```

Zadanie 11.

Znajdź błędy w poniższym fragmencie programu:

```
const double g=16.;
double t;
cout << "Wpisz czas w sek. ";
cin >> t;
cout << "Droga " << 1/2*(g*t*t);
```

Zadanie 12.

Uprość wyrażenie:

```
bool wewnatrz = !((x<lewa) || (x>prawa) || (y<gora) || (y >dol));
```

Zadanie 13.

Jakiemu wyrażeniu jest równoważne wyrażenie `!((x > y) && (y <= 3))` ?

- A. `(x > y) && (y <= 3)`
- B. `(x > y) || (y <= 3)`
- C. `(x < y) || (y >= 3)`
- D. `(x <= y) || (y > 3)`
- E. `(x <= y) && (y > 3)`

Zadanie 14

Kiedy wartość wyrażenia `!(a && b) && (a || b)` wynosi `true` ?

- A. Zawsze
- B. Nigdy
- C. Kiedy `a` i `b` są równe `true`
- D. Kiedy ani `a`, ani `b` nie jest równe `true`
- E. Kiedy tylko `a`, albo tylko `b` jest równe `true`

Zadanie 15.

Napisz funkcję, która zwraca odległość pomiędzy punktami $(x1, y1)$ i $(x2, y2)$. Wzór na odległość to:

$$d = \sqrt{(x2-x1)^2 + (y2-y1)^2}$$

Funkcja ma następujący prototyp:

```
double Odl(int x1, int y1, int x2, int y2);
```

Do wyznaczenia pierwiastka użyj funkcji bibliotecznej

```
double sqrt(double s),
```

która wymaga pliku nagłówkowego

```
#include <cmath>
```

(w starszych wersjach kompilatora math.h).

Zadanie 16.

Napisz funkcję, która zamienia zużycie benzyny w litrach na sto kilometrów na zużycie w milach na galon (to nie jest pomyłka!). 100 km to 62.14 mili, 1 galon to 3.785 litra. Zatem 19 mpg (mil na galon) to około 12.4 l/100 km. Napisz program, który prosi użytkownika o podanie zużycia benzyny w l/km i wyświetla w mpg.