

--

1	2	3	4	Σ

Zadanie 1: Zakładając istnienie struktury `Para` wyposażonej w odpowiednie składowe wynikające z poniższych kodów, przeanalizuj działanie (lub niedziałanie) następujących funkcji `MM1`, `MM2`, `MM3`:

```
double MM1(Para p, Para q) { return fabs(p.x - q.x) + fabs(p.y - q.y); }
double MM2(Para &p, Para &q) { return fabs(p.x - q.x) + fabs(p.y - q.y); }
double MM3(const Para &p, const Para &q) { return fabs(p.x - q.x) + fabs(p.y - q.y); }
```

zamiast funkcji `MM` w kodzie:

```
int Najlepszy(Para *pary, int ile) {
    double naj = std::numeric_limits<double>::infinity(), z;
    int naji = -1;
    for (int i=1; i<ile-1; i++) {
        z = MM(pary[i], pary[i-1] + pary[i+1]);
        if (z < naj) { naj = z; naji = i; }
    }
    return naji;
}
```

Wyjaśnij różnice, zwłaszcza w kontekście l-wartości i r-wartości oraz ukrytych wywołań konstruktora kopiującego. Napisz krótko, jednym zdaniem, co wyznacza funkcja `Najlepszy`.

Zadanie 2: Napisz klasę `Napis` ułatwiającą korzystanie z ciągów znaków (jak `string`, ale bez wykorzystania klasy `string`), posiadającą pole z napisem reprezentowanym jako `char *`, konstruktor bezargumentowy, konstruktor kopiujący, destruktory, oraz operator `+=` dodający na koniec napisu inny napis i operator `+` zwracający takie połączenie napisów. Niech operator `+=` będzie metodą klasy `Napis` a operator `+` funkcją zewnętrzną. Stwórz również niezbędne składowe, które pozwolą na konwersję obiektu typu `Napis` na `char *` i odwrotnie. Skomentuj rozwiązanie.

Zadanie 3: Napisz klasy `Powiadamiacz`, jej potomne `PowWKonsoli` i `PowMailem`, oraz klasę `LiczIInformuj` z kolekcją powiadamiaczy (może być `vector` lub zwykła tablica) i metodą `Licz()`, która przy pierwszym, dziesiątym, setnym, itd. (każdym będącym potęgą liczby 10) wywołaniu zawoła odpowiednie metody `Powiadom()` wszystkich powiadamiaczy z kolekcji. Niech funkcja `Powiadom()` ma argument będący tekstem powiadomienia. Klasa `PowWKonsoli` niech powiadamia przez wysłanie komunikatu na standardowe wyjście, a `PowMailem` niech wysyła komunikat na określony w konstruktorze adres email (można nie pisać funkcji wysyłającej email ;-)).

Zadanie 4: Napisz szablon funkcji, która dwie posortowane nierosnąco tablice (elementów typu będącego parametrem szablonu) połączy w jedną tablicę, posortowaną niemalejąco (zaalokuje, wypełni i zwróci).
