Lab nr I

1. Zadeklarować zmienną typu `int`, przypisać jej wartość (inicjalizacja lub poprzez przypisanie), wyświetlić jej wartość na standardowe wyjście, za pomocą `cout` oraz funkcji `printf`.
2. Zadeklarować funkcję `int liczSumuj()` oraz ją zdefiniować tak, aby wyświetlała 100 liczb typu integer na standardowe wyjście i ostatecznie zwracała sumę tych liczb. Użyć tej funkcji.
3. Wyświetlić adres zmiennej z punktu nr 1 na standardowe wyjście (np. poprzez `cout`).
4. Zadeklarować zmienną "dynamiczną" `int` (**wskaźnik** na `int`), ulokować dla niej obszar w pamięci (`new`), przypisać jej wartość. Wyświetlić na standardowe wyjście jej adres, a potem jej wartość. Zmienić jej wartość, np. odjąć 1 i ponownie wyświetlić adres i potem wartość. **Na koniec programu dealokować pamięć** (`delete`).
5. Zadeklarować klasę `Punkt`, reprezentującą punkt na płaszczyźnie, zawierającą dwa publiczne (`public:`)pola typu `int` o nazwach `x` i `y`. Zadeklarować i użyć (zapisać wartość i odczytać wartość) obiekt klasy `Punkt`.
6. Tak jak 5, ale z użyciem `struct`, klasa tutaj niech ma inną nazwę, ale oprócz tych 2 zmian, jej działanie ma być indentyczne do tego z 5.
7. Stworzyć, użyć (zapisać wartość i odczytać wartość) i zniszczyć obiekt "dynamiczny" klasy `Punkt` (**wskaźnik** na `Punkt`, użycie to tutaj: zapisywanie i odczytywanie wartości z pól `x` i `y`, dealokacja obiektu za pomocą operatora `delete`).
8. Do klasy `Punkt` dopisać funkcję klasową (tzw. metodę) `void wypisz()`, która będzie wypisywać w konsoli zawartość obiektu, np. `"Punkt (3;4)"`. Użyć tej metody.
9. Do klasy `Punkt` dopisać metodę `double odlegloscZero()`, która będzie zwracać odległość punktu od środka układu współrzędnych. Do obliczeń wykorzystać funkcję `double sqrt(double)` z biblioteki `math.h`, która zwraca pierwiastek kwadratowy.
10. Do klasy `Punkt` dopisać metodę `double odlegloscOdPunktu(Punkt pp)`, która będzie zwracać odległość punktu, dla którego wywołano metodę od punktu podanego jako parametr. Użyć tej metody.
11. **(!)** Do klasy `Punkt` dopisać metodę `double odlegloscOdPunktu(Punkt* Pointer_pp)`, która będzie zwracać odległość punktu, dla którego wywołano metodę od punktu podanego jako parametr. Użyć tej metody.
12. Stworzyć tablicę 10 liczb `int`, wypełnić przykładowymi liczbami, policzyć sumę i wyświetlić ją na standardowe wyjście.
13. **(!)** Pobrać od użytkownika liczbę elementów `N` (`int`, zakładamy, że większy od zera), zadeklarować i ulokować tablicę dynamiczną przechowującą `N` liczb typu `int`, wypełnić ją wartościami od 1 do `N`. Dealokować pamięć!
14. **(!)** Stworzyć tablicę (może być statyczna) 5 obiektów klasy `Punkt`, wypełnić ich współrzędne wartościami kolejno: (0,4); (1,3); (2,2); (3,1); (4,0). Policzyć (w pętli (podwójnej?)) i wyświetlić odległości pomiędzy każdym z osobnych punktów w tablicy. Jak udało się zrobić punkt 11: spróbować podać jako argument wskaźnik do danego punktu.