

1. Proszę napisać definicję klasy `Zesp` reprezentującej liczby zespolone w postaci dwóch pól typu `double`. Definicja klasy powinna zawierać:

- Dwa pola reprezentujące część rzeczywistą i urojoną liczby.
- Konstruktor bez argumentów (domyślny) inicjujący każde z pól wartością 0.
- Konstruktor z jednym argumentem typu `double` inicjujący część rzeczywistą liczby wartością argumentu, a część urojoną liczby – wartością 0.
- Konstruktor z dwoma argumentami typu `double` inicjujący część rzeczywistą i urojoną liczby wartościami argumentów.
- Metodę `abs()` zwracającą wartość bezwzględną liczby (jako wartość typu `double`).
- Operatory dodawania, mnożenia, odejmowania i dzielenia oraz odpowiadające im operatory modyfikacji implementujące odpowiednie operacje w sposób właściwy dla liczb zespolonych.

Proszę napisać program obliczający wartości bezwzględne sumy, iloczynu, różnicy oraz ilorazu dwóch liczb zespolonych, dla których części rzeczywiste i urojone będą wprowadzane przez użytkownika jako dane wejściowe, z wykorzystaniem klasy `Zesp`.

2. Proszę napisać definicję klasy `Matrix2x2` reprezentującej macierze 2 x 2 w postaci czterech pól typu `double`. Definicja klasy powinna zawierać:

- Cztery pola reprezentujące elementy macierzowe.
- Konstruktor bez argumentów (domyślny) inicjujący każde z pól wartością 0.
- Konstruktor z jednym argumentem typu `double` inicjujący diagonalne elementy macierzowe wartością argumentu, a pozadiagonalne elementy macierzowe – wartością 0.
- Konstruktor z czterema argumentami typu `double` inicjujący elementy macierzowe wartościami argumentów.
- Metodę `det()` zwracającą wartość wyznacznika macierzy (jako wartość typu `double`).
- Operatory dodawania, mnożenia i odejmowania oraz odpowiadające im operatory modyfikacji implementujące odpowiednie operacje w sposób właściwy dla macierzy 2 x 2.

Proszę napisać program obliczający wyznaczniki sumy, iloczynu i różnicy dwóch macierzy 2 x 2, dla których elementy macierzowe będą wprowadzane przez użytkownika jako dane wejściowe, z wykorzystaniem klasy `Matrix2x2`.

3. Proszę dodać do klasy `Matrix2x2` z poprzedniego zadania operator `<<`, pozwalający na zapisanie elementów macierzowych macierzy na strumień reprezentowany przez obiekt klasy `ostream`. Operator ten powinien być tak zdefiniowany, aby można było łączyć reprezentowaną przez niego operację z innymi operacjami wyjścia na ten sam strumień.

4. Proszę zdefiniować klasę `Resistor` do reprezentowania oporników o zadanych oporach. Proszę zaimplementować:

- Konstruktor jednoargumentowy, który inicjalizuje opór wartością swojego argumentu.
- Operatory `+` i `-` obliczające opór odpowiednio przy równoległym i szeregowym łączeniu oporników.
- Odpowiadające powyższym operatorom operatory `+=` i `-=`.
- Operator `<` porównujący dwa opory.
- Operator `<<` zapisujący opór do strumienia typu `ostream` oraz operator `>>` wczytujący opór ze strumienia typu `istream`.

Proszę napisać program `resistor` wczytujący ze standardowego wejścia opory dwóch oporników i wyprowadzający na standardowe wyjście opory uzyskane przez ich połączenie równoległe oraz szeregowe oraz używający operatora porównania zdefiniowanego dla klasy `Resistor` do określenia który z tych oporów jest większy.

5. Dane są dwa pliki, `auta.txt` oraz `motocykle.txt`, zawierające informacje o pewnej liczbie pojazdów. Każdy wiersz pliku `auta.txt` zawiera:

- markę (ciąg znaków),
- prędkość maksymalną w km/h (liczba całkowita),
- maksymalną liczbę pasażerów (łącznie z kierowcą),
- liczbę drzwi,
- kolor

samochodu, rozdzielone znakami przerwy. Każdy wiersz pliku `motocykle.txt` zawiera:

- markę (ciąg znaków),
- prędkość maksymalną w km/h (liczba całkowita),
- informację o tym, czy może jeździć w terenie (0 lub 1),
- kolor

dla motocykla, rozdzielone znakami przerwy. Proszę napisać program, który znajdzie pojazd o największej prędkości maksymalnej oraz wyznaczy liczbę pojazdów w kolorze czerwonym w obu plikach. Program proszę zaprojektować w oparciu o klasy. Proszę zdefiniować klasę `Pojazd` zawierającą pola:

- `marka` (`string`),
- `pr_maks` (`unsigned int`),
- `kolor` (`string`)

oraz metody: `kolor_czerwony()`, zwracającą `true` dla pojazdów w kolorze czerwonym (i `false` dla pozostałych), `prn()`, zwracającą wartość pola `pr_maks` oraz `mar()`, zwracającą wartość pola `marka`. Ponadto proszę zdefiniować klasy `Auto` oraz `Motocykl`, pochodne w stosunku do `Pojazd`, i dla każdej z nich operator `>` pozwalający na wczytanie jednego wiersza z pliku `auta.txt` lub `motocykle.txt`, odpowiednio, do obiektu danej klasy. Proszę wykorzystać wymienione klasy, metody i operatory do uzyskania wyniku.

6. Proszę napisać i skompilować program wyświetlający napis „Hello world!” w oddzielnym oknie z wykorzystaniem biblioteki `Qt` (kod źródłowy takiego programu znajduje się w II części prezentacji do wykładu).
7. Proszę napisać program rysujący wielokąt foremny o  $n$  wierzchołkach w oddzielnym oknie z wykorzystaniem biblioteki `Qt`. Rysunek powinien być umieszczony w centralnej części okna i jego rozmiary powinny zmieniać się odpowiednio w stosunku do bieżących rozmiarów okna. Liczba  $n$  powinna być wczytywana ze standardowego wejścia przed otwarciem okna z rysunkiem.
8. Proszę napisać program rysujący wielokąt foremny o  $n$  wierzchołkach, obrócony o  $s$  stopni w kierunku zgodnym z ruchem wskazówek zegara w stosunku do pionowej osi symetrii ekranu, w oddzielnym oknie z wykorzystaniem biblioteki `Qt`. Rysunek powinien być umieszczony w centralnej części okna i jego rozmiary powinny zmieniać się odpowiednio w stosunku do bieżących rozmiarów okna. Liczby  $n$  oraz  $s$  powinny być wczytywane ze standardowego wejścia przed otwarciem okna z rysunkiem.
9. Proszę napisać program rysujący  $k$  wielokątów foremnych o  $n$  wierzchołkach tak, aby
- wszystkie wielokąty miały boki o jednakowej długości,
  - ich środki symetrii pokrywały się,
  - każdy wielokąt był obrócony w stosunku do poprzedniego o kąt  $2\pi/k$ ,

w oddzielnym oknie z wykorzystaniem biblioteki `Qt`. Rysunek powinien być umieszczony w centralnej części okna i jego rozmiary powinny zmieniać się odpowiednio w stosunku do bieżących rozmiarów okna. Liczby  $k$  i  $n$  powinny być wczytywane ze standardowego wejścia przed otwarciem okna z rysunkiem.

10. Wykorzystując bibliotekę `Qt` proszę napisać program rysujący  $k$  okręgów o środkach rozmieszczonych symetrycznie na okręgu o  $n$ -krotnie większym promieniu i środku pokrywającym się ze środkiem dostępnego okna. Rozmiary rysunku powinny zmieniać się odpowiednio w stosunku do bieżących rozmiarów okna. Liczby  $k$  i  $n$  powinny być wczytywane ze standardowego wejścia przed otwarciem okna z rysunkiem. **Wskazówka:** Do rysowania okręgów należy użyć jednego z wariantów metody `drawEllipse()` z klasy `QPainter`.
11. Wykorzystując bibliotekę `Qt` proszę napisać program rysujący figurę, jak w zadaniu 7, ale tak, aby wartość  $n$  była wprowadzana do programu z pomocą pola edycyjnego klasy `QLineEdit` i zatwierdzana z pomocą przycisku klasy `QPushButton`.

12. Wykorzystując bibliotekę *Qt* proszę napisać program rysujący figurę, jak w zadaniu 9, ale tak, aby wartości  $n$  i  $k$  były wprowadzane do programu z pomocą pól edycyjnych klasy `QLineEdit` i zatwierdzane z pomocą przycisku klasy `QPushButton`. Proszę dodać do interfejsu użytkownika przycisk klasy `QPushButton`, którego naciśnięcie będzie powodować zakończenie działania programu.
13. Wykorzystując bibliotekę *Qt* proszę napisać program rysujący  $k$  przystających wielokątów foremnych o  $m$  wierzchołkach i środkach symetrii rozmieszczonych w jednakowych odstępach na okręgu o promieniu  $n$ -krotnie większym od promienia okręgu opisanego na jednym z tych wielokątów i środku pokrywającym się ze środkiem dostępnego okna. Każdy wielokąt powinien być obrócony o kąt  $2\pi/k$  w stosunku do poprzedniego. Rozmiary rysunku powinny zmieniać się odpowiednio w stosunku do bieżących rozmiarów okna. Liczby  $k$ ,  $m$  i  $n$  powinny być wprowadzane do programu z pomocą komponentów klasy `QSpinBox` i zatwierdzane z pomocą przycisku klasy `QPushButton`. Proszę dodać do interfejsu użytkownika przycisk klasy `QPushButton`, którego naciśnięcie będzie powodować zakończenie działania programu.
14. Wykorzystując bibliotekę *Qt* proszę napisać program rysujący tor dla dwuwymiarowego tłumionego oscylatora harmonicznego o masie  $m$  i współczynniku tarcia  $\gamma$ . Masa oscylatora, współczynnik tarcia oraz współrzędne położenia początkowego i prędkości początkowej powinny być wprowadzane do programu z pomocą graficznego interfejsu użytkownika.
15. Katapulta o wysokości  $h$  wyrzuca pod kątem  $\alpha$  do powierzchni ziemi pocisk o masie  $m$  poruszający się w powietrzu, które stawia opór proporcjonalny do prędkości pocisku, przy czym współczynnik tego oporu ma wartość  $\gamma$ . Proszę napisać program obliczający prędkość początkową, z jaką należy wystrzelić pocisk, aby osiągnął on najwyższą wysokość nad powierzchnią ziemi w zadanej odległości  $l$  od punktu, w którym znajduje się katapulta. Program powinien podawać także tę wysokość oraz czas, po jakim zostanie ona osiągnięta od chwili wystrzału. Liczby  $h$ ,  $\alpha$ ,  $m$ ,  $\gamma$  i  $l$  powinny być danymi wejściowymi dla programu (mogą one być wprowadzane do programu z pomocą graficznego interfejsu użytkownika). Można przyjąć, że przyspieszenie ziemskie ma wartość  $g = 9,81 \text{ m/s}^2$ .
16. Proszę napisać program rysujący tor cząstki o masie  $m$  i ładunku elektrycznym  $e$ , poruszającej się w skrzyżowanych polach elektrycznym i magnetycznym, które są stałe, jednorodne i wzajemnie prostopadłe, w ośrodku z tarciem o współczynniku  $\gamma$ , przy założeniu, że ruch odbywa się w płaszczyźnie prostopadłej do pola magnetycznego.