

1. Proszę napisać program znajdujący w pliku, którego nazwa jest jego (pierwszym) argumentem wszystkie wiersze tekstu zawierające podany wzorzec, będący (drugim) argumentem programu.
2. Proszę napisać program, który w każdym wierszu odczytanym z pliku, będącym jego (pierwszym) argumentem, zamienia wzorzec, będący jego drugim argumentem, (o ile występuje on w tym wierszu) na ciąg znaków będący trzecim argumentem programu.
3. Proszę napisać program o 4 argumentach, z których pierwszy jest nazwą pliku do przeszukania, drugi – ciągiem znaków identyfikującym wiersze do przetworzenia, a trzeci i czwarty – wzorcem oraz ciągiem znaków mającym zastąpić wzorzec, odpowiednio. Program powinien znaleźć w pliku o podanej nazwie wszystkie wiersze, w których występuje podany ciąg znaków (drugi argument), a następnie *w tych wierszach* zamienić wzorzec (trzeci argument) na podany substytut (czwarty argument).
4. 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`.

5. 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`.

6. 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.
7. 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.

8. 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), `prm()`, 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.

9. Dane są dwa pliki, `koty.txt` oraz `psy.txt`, zawierające informacje o pewnej liczbie zwierząt domowych. Każdy wiersz pliku `koty.txt` zawiera:

- imię zwierzaka (ciąg znaków),
- nazwę rasy (ciąg znaków),
- masę ciała w kilogramach (liczba całkowita),
- długość w centymetrach (liczba całkowita),
- umaszczenie (ciąg znaków)

dla każdego kota, rozdzielone znakami przerwy. Każdy wiersz pliku `psy.txt` zawiera:

- imię zwierzaka (ciąg znaków),
- nazwę rasy (ciąg znaków),
- masę ciała w kilogramach (liczba całkowita),
- długość w centymetrach (liczba całkowita),
- wzrost w centymetrach (liczba całkowita),
- umaszczenie (ciąg znaków)

dla każdego psa, rozdzielone znakami przerwy. Proszę napisać program, który znajdzie zwierzęta o największej i najmniejszej masie ciała oraz wyznaczy liczbę zwierząt o czarnym umaszczeniu w obu plikach. Program proszę zaprojektować w oparciu o klasy. Proszę zdefiniować klasę `Zwierzak` zawierającą pola:

- `imie` (`string`),
- `rasa` (`string`),
- `waga` (`unsigned int`),
- `kolor` (`string`)

oraz metody: `kolor_czarny()`, zwracającą `true` dla zwierząt o czarnym umaszczeniu (i `false` dla pozostałych), `waga()`, zwracającą wartość pola `waga` oraz `rasa()`, zwracającą wartość pola `rasa`. Ponadto proszę zdefiniować klasy `Kot` oraz `Pies`, pochodne w stosunku do `Zwierzak`, i dla każdej z nich operator `>>` pozwalający na wczytanie jednego wiersza z pliku `koty.txt` lub `psy.txt`, odpowiednio, do obiektu danej klasy. Proszę wykorzystać wymienione klasy, metody i operatory do uzyskania wyniku.

10. 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).
11. 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.
12. 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.
13. 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.

14. 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`.
15. Wykorzystując bibliotekę `Qt` proszę napisać program rysujący figurę, jak w zadaniu 11, ale tak, aby wartość n była wprowadzana do programu z pomocą pola edycyjnego klasy `QLineEdit` i zatwierdzana z pomocą przycisku klasy `QPushButton`.
16. Wykorzystując bibliotekę `Qt` proszę napisać program rysujący figurę, jak w zadaniu 13, 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.
17. Proszę napisać program rysujący, z wykorzystaniem biblioteki `Qt`, 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.
18. Proszę napisać program rysujący, z wykorzystaniem biblioteki `Qt`, wykres funkcji danej w postaci pliku tekstowego, zawierającego w każdym wierszu parę liczb: wartość zmiennej niezależnej oraz odpowiadającą jej wartość funkcji. Nazwa pliku powinna być argumentem programu.
19. Proszę napisać program rysujący, z wykorzystaniem biblioteki `Qt`, wykres funkcji jak w zadaniu 18 wraz z podziałką z odpowiednimi wartościami liczbowymi na osiach.
20. Proszę napisać program rysujący, z wykorzystaniem biblioteki `Qt`, wykres funkcji jak w zadaniu 19, tylko dla nazwy pliku wprowadzanej do programu z pomocą graficznego interfejsu użytkownika. Nazwa pliku z danymi powinna być wprowadzana do programu z pomocą pola edycyjnego `QLineEdit`, a wykres powinien być rysowany po naciśnięciu przez użytkownika przycisku klasy `QPushButton`. Ponadto do interfejsu użytkownika proszę dodać przycisk klasy `QPushButton`, którego naciśnięcie będzie powodować zakończenie działania programu.

21. Do interfejsu użytkownika w programie z zadania 20 proszę dodać komponenty pozwalające użytkownikowi na określenie liczby podziałek na osiach oraz zakresów wartości dla osi \hat{x} i \hat{y} .
22. Proszę napisać program wyznaczający ruch jednowymiarowego tłumionego oscylatora harmonicznego z użyciem algorytmu Rungego-Kutty (czwartego rzędu). Wyniki działania programu powinny być zapisywane w pliku będącym jego argumentem w taki sposób, aby w każdym wierszu pliku znajdowały się trzy liczby, reprezentujące chwilę czasu oraz położenie i prędkość oscylatora w tej chwili czasu, odpowiednio.
23. Proszę napisać program wyznaczający ruch jednowymiarowego tłumionego oscylatora harmonicznego z użyciem algorytmu Rungego-Kutty (czwartego rzędu) i rysujący wykres położenia w funkcji czasu dla tego oscylatora (bez podziałki).
24. Proszę napisać program wyznaczający ruch jednowymiarowego tłumionego oscylatora harmonicznego z użyciem algorytmu Rungego-Kutty (czwartego rzędu) i rysujący wykresy położenia i prędkości w funkcji czasu dla tego oscylatora (bez podziałki).
25. Proszę napisać program wyznaczający ruch dwuwymiarowego tłumionego oscylatora harmonicznego z użyciem algorytmu Rungego-Kutty (czwartego rzędu) i rysujący tor ruchu tego oscylatora (bez podziałki).
26. Proszę napisać program rysujący, z wykorzystaniem biblioteki Qt , wykres położenia w funkcji czasu dla tłumionego oscylatora harmonicznego o masie m i współczynniku tarcia γ . Parametry oscylatora oraz współrzędne położenia początkowego i prędkości początkowej powinny być wprowadzane do programu z pomocą graficznego interfejsu użytkownika. Wykres powinien być rysowany z podziałką.