Zadanie 1

Napisz klasę Letter, która będzie służyć do analizowania tekstów. Obiekty klasy zawierają tablicę 26 liczb typu int odpowiadających literom alfabetu angielskiego (element o indeksie 0 — literze 'A', ..., o indeksie 25 — literze 'Z'). Konstruktor klasy pobiera tekst (jako wskaźnik typu const char*) i wypełnia tablicę liczbami wystąpień poszczególnych liter.

Uwaga: litery duże i odpowiadające im małe są uważane za takie same.

Klasa udostępnia metody

- getMostFrequent zwracającą znak, który występuje najczęściej;
- numOfDifferent zwracającą liczbę różnych liter;
- printFrequencies drukującą "histogram" opisujący liczby wystąpień poszczególnych liter: wyrównany od dołu, z kolumnami złożonymi z tylu gwiazdek, ile wynosiła liczba wystąpień odpowiedniej litery (patrz przykład poniżej).

Na przykład następujący program

```
download LetterHist.cpp
#include <iostream>
class Letters {
    int letters[26]{0};
    // ...
};
int main() {
    const char* text =
             "To be, or not to be- that is the questiona\n:"
             "Whether 'tis nobler in the mind to suffer\n"
             "The slings and arrows of outrageous fortune\n"
             "Or to take arms against a sea of troubles, \n"
             "And by opposing end them.";
    Letters lett(text);
    std::cout << "Most frequent letter:</pre>
               << lett.getMostFrequent() << std::endl;</pre>
    std::cout << "Number of different letters: "</pre>
               << lett.numOfDifferent() << std::endl;
    std::cout << "Frequency table of all letters:\n";</pre>
    std::cout << std::endl;</pre>
    lett.printFrequencies();
}
```

powinien wydrukować

Most frequent letter: T Number of different letters: 21 Frequency table of all letters:

				*		
			*	*		
	*		*	*		
	*		*	*		
	*		*	*		
	*		*	*		
	*		*	**		
*	*		**	**		
*	*		**	***		
*	*		**	***		
*	*		**	***		
*	*	*	**	***		
*	*	**	**	***		
*	*	**	**	****		
**	**	**	**	****		
**	***	* **	**	****		
**	***	* **	****	****		
**	***	***	****	****	*	
**	***	* **	*****	****	*	*
ABCDEFGHIJKLMNOPQRSTUVWXYZ						

Zadanie 2 _

Napisz i przetestuj funkcję pobierającą

- 1. tablice wskaźników do funkcji typu double → double,
- 2. jej wymiar,
- 3. dwie liczby typu **double** definiujące przedział [a, b],
- 4. adres istniejącej zmiennej typu double.

a zwracającą wskaźnik do funkcji, również typu double \rightarrow double.

Funkcja oblicza dla każdej z funkcji wskazywanych przez elementy tablicy jej maksymalną wartość na odcinku [a,b] i zwraca wskaźnik do tej z funkcji, dla której to maksimum wypadnie najwieksze.

Aby znaleźć maksimum funkcji na odcinku, można "przejechać" przez ten odcinek z małym krokiem (np. $\epsilon=10^{-4}$) i znajdować w każdym punkcie wartość funkcji.

Do zmiennej wskazywanej przez ostatni argument (pxmax) należy wpisać wartość argumentu (czyli "iksa"), dla którego znaleziona funkcja miała największą wartość.

W programie testowym można użyć wskaźników do funkcji bibliotecznych z nagłówka *cmath* (jak sin, cos, exp) oraz przynajmniej jednej funkcji napisanej samemu.

Na przykład program

```
#include <iostream>
#include <cmath>
```

```
using namespace std;
    double f1(double x) { return x/4; }
    double f2(double x) { return -2*x; }
    using D2D = double(*)(double);
    D2D maxfun(D2D funs[], int size,
            double a, double b, double* pmax) {
        static const double eps = 1e-4;
        // ...
    }
    int main() {
        double xmax;
        D2D funs[] = {f1,sin,f2};
        D2D pf = maxfun(funs,3,0,3,&xmax);
        cout << "xmax = " << xmax << "; f(xmax) = "</pre>
            << pf(xmax) << endl;</pre>
    }
powinien wydrukować
    xmax = 1.5708; f(xmax) = 1
(bo z trzech funkcji f1, sin i f2 na odcinku [0,3] największą wartość, równą 1, przyj-
muje druga z nich dla argumentu \pi/2).
```