

Podstawy Programowania Komputerów

Strumienie

7 listopada 2018

```
/** Funkcja zapisuje do strumienia przekazane liczby  
@param str strumień do zapisu  
@param liczby liczby do zapisu do strumienia */  
void zapisz_liczby (std::ostream & str, const std::vector<double> & liczby);
```

```
/** Funkcja zapisuje do podanego strumienia losową liczbę losowych liczb  
zmiennoprzecinkowych */  
void zapisz_losowe (std::ostream & str);
```

```
/** Funkcja zwraca średnią liczby podanych w strumieniu.  
@param str strumień z liczbami  
@return średnia liczb podanych w strumieniu, gdy nie ma żadnej liczby w  
strumieniu, funkcja zwraca 0 */  
double srednia_liczb (std::istream & str);
```

```
/** Funkcja wyznacza normę Frobeniusa podanej macierzy (tablicy  
dwuwymiarowej), czyli sumę kwadratów wszystkich wartości.  
@param str strumień z macierzą  
@return norma Frobeniusa */  
double frobenius (std::istream & str);
```

```
/** Funkcja usuwa duplikaty liczb całkowitych z wejścia. Jako duplikat  
rozumiemy następujące bezpośrednio po sobie takie same liczby. W  
przypadku duplikatów do wyjścia zostanie zapisane tylko jedno wystąpienie  
zamiast kilku.  
@param wejscie strumień wejściowy  
@param wyjscie strumień wyjściowy  
*/  
void uniq (std::istream & wejscie, std::ostream & wyjscie);
```

```
/** Funkcja zlicza, ile jest liczb w każdym wierszu przekazanym przez  
strumień wejściowy. Liczba liczb w wierszu jest wypisywana do strumienia  
wyjściowego.  
@param wejscie strumień wejściowy  
@param wyjscie strumień wyjściowy  
*/  
void ile_w_wierszu (std::istream & wejscie, std::ostream & wyjscie);
```

```
/** Funkcja wyznacza średnią liczb (w każdym wierszu osobno) w strumieniu  
wejściowym. Średnia liczb jest zapisywana do strumienia wyjściowego.  
@param wejscie strumień wejściowy
```

```

@param wyjście strumień wyjściowy
*/
void srednia_wiersza (std::istream & wejscie, std::ostream & wyjscie);



---


/** Funkcja transponuje macierz (tablicę dwuwymiarową) liczb
    zmiennoprzecinkowych odczytaną ze strumienia wejściowego. Transponowana
    macierz zostaje zapisana do strumienia wynikowego.
    @param wejscie strumień wejściowy
    @param wyjście strumień wyjściowy
    */
void transponuj (std::istream & wejscie, std::ostream & wyjscie);



---


/** Funkcja zapisuje do strumienia wyjściowego pierwsze słowo z każdego
    wiersza tekstu przekazanego przez strumień wejściowy.
    @param wejscie strumień wejściowy
    @param wyjście strumień wyjściowy
    */
void pierwsze (std::istream & wejscie, std::ostream & wyjscie);



---


/** Funkcja rozdziela podany tekst na poszczególne słowa. Dwa słowa w teks-
    cie rozdzielone są minimalnie jednym białym znakiem.
    @param tekst do rozdzielenia
    @return wektor słów tekstu
    */
std::vector<std::string> rozdziel (const std::string & tekst);



---


/** Funkcja tworzy histogram częstości występowania liter (wielkie i małe
    litery nie są rozróżniane) w tekście.
    @param nazwa_pliku nazwa pliku wejściowego z tekstem
    @return wektor licznosci liter od a do z w tekście z pliku wejściowego
    */
std::vector<int> tworz_histogram (const std::string & nazwa_pliku);



---


/** Funkcja zapisuje histogram do pliku. Dla każdej litery wypisywany jest
    bloczek histogramu zbudowany z gwiazdek, przy czym maksymalna liczba
    gwiazdek jest podana jako parametr funkcji. Tę wartość gwiazdek ma najczę-
    stsza litera, bloczki dla pozostałych liter mają proporcjonalną wysokość.
    @param histogram histogram liczebności liter
    @param nazwa_pliku nazwa pliku do zapisania histogramu
    @param maksimum maksymalna wysokość bloczka w histogramie, pozostałe są
        liniowo proporcjonalne
    */
void zapisz_histogram (const std::vector<int> & histogram, const std::string
    & nazwa_pliku, int maksimum = 30);

```