

# Kalkulator

v1

Generated by Doxygen 1.9.6



---

<b>1 File Index</b>	<b>1</b>
1.1 File List	1
<b>2 File Documentation</b>	<b>3</b>
2.1 header.h File Reference	3
2.1.1 Function Documentation	3
2.1.1.1 dziesiętna_bin()	3
2.1.1.2 dziesiętna_hex()	4
2.1.1.3 dziesiętna_oct()	4
2.1.1.4 konwersja_na_postfiks()	4
2.1.1.5 oblicz_wyrażenie()	5
2.1.1.6 rozbicie()	5
2.1.1.7 wczytaj_pamięć()	6
2.1.1.8 zapisz_pamięć()	6
2.2 header.h	7
2.3 main.cpp File Reference	7
2.3.1 Function Documentation	8
2.3.1.1 dziesiętna_bin()	8
2.3.1.2 dziesiętna_hex()	8
2.3.1.3 dziesiętna_oct()	8
2.3.1.4 konwersja_na_postfiks()	9
2.3.1.5 main()	9
2.3.1.6 oblicz_wyrażenie()	9
2.3.1.7 rozbicie()	10
2.3.1.8 wczytaj_pamięć()	10
2.3.1.9 zapisz_pamięć()	11
<b>Index</b>	<b>13</b>



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">header.h</a>	.....	<a href="#">3</a>
<a href="#">main.cpp</a>	.....	<a href="#">7</a>



## Chapter 2

# File Documentation

## 2.1 header.h File Reference

### Functions

- `vector< string > rozbiecie` (`string &wyrażenie`, `unordered_map< string, int > &operacja`)  
*Funkcja rozbijająca podane wyrażenie na pojedyncze symbole.*
- `vector< string > konwersja\_na\_postfiks` (`vector< string > zapis_infiksowy`, `unordered_map< string, int > &operacja`)  
*Funkcja zamieniająca zapis infiksowy wyrażenia na zapis postfiksowy.*
- `float oblicz\_wyrażenie` (`vector< string > &zapis_postfiksowy`, `unordered_map< string, int > &operacja`, `string wartosci_pamieci[]`)  
*Funkcja obliczająca wyrażenie z zapisu postfiksowego.*
- `void dziesiętna\_bin` (`int &całkowita_int`, `float &ulamkowa`, `const int &precyzja`)  
*Funkcja konwertująca liczbę dziesiętną na binarną.*
- `void dziesiętna\_hex` (`int &całkowita_int`, `float &ulamkowa`, `const int &precyzja`)  
*Funkcja konwertująca liczbę dziesiętną na szesnastkową.*
- `void dziesiętna\_oct` (`int &całkowita_int`, `float &ulamkowa`, `const int &precyzja`)  
*Funkcja konwertująca liczbę dziesiętną na osemkową.*
- `void zapisz\_pamięć` (`string &nazwa_pliku`, `const int &N`, `string wartosci_pamieci[]`, `bool &ui_plik`)  
*Funkcja zapisująca wartości pamięci do pliku.*
- `void wczytaj\_pamięć` (`string &nazwa_pliku`, `const int &N`, `string wartosci_pamieci[]`, `bool &ui_plik`)  
*Funkcja wczytująca wartości pamięci z pliku.*

### 2.1.1 Function Documentation

#### 2.1.1.1 `dziesiętna_bin()`

```
void dziesiętna_bin (
    int & całkowita_int,
    float & ulamkowa,
    const int & precyzja )
```

Funkcja konwertująca liczbę dziesiętną na binarną.

Funkcja pobiera osobno część całkowitą i ułamkową wyniku, a następnie zamienia je w zapis binarny

## Parameters

<i>calkowita_int</i>	Czesc calkowita wyniku wyrazenia
<i>ulamkowa</i>	Czesc ulamkowa wyniku wyrazenia
<i>precyzja</i>	Stala decydujaca o ilosci cyfr po przecinku

**2.1.1.2 dziesiętna\_hex()**

```
void dziesiętna_hex (
    int & calkowita_int,
    float & ulamkowa,
    const int & precyzja )
```

Funkcja konwertująca liczbę dziesiętną na szesnastkową.

Funkcja pobiera osobno część całkowitą i ułamkową wyniku, a następnie zamienia je w zapis szesnastkowy

## Parameters

<i>calkowita_int</i>	Czesc calkowita wyniku wyrazenia
<i>ulamkowa</i>	Czesc ulamkowa wyniku wyrazenia
<i>precyzja</i>	Stala decydujaca o ilosci cyfr po przecinku

**2.1.1.3 dziesiętna\_oct()**

```
void dziesiętna_oct (
    int & calkowita_int,
    float & ulamkowa,
    const int & precyzja )
```

Funkcja konwertująca liczbę dziesiętną na ośmiową.

Funkcja pobiera osobno część całkowitą i ułamkową wyniku, a następnie zamienia je w zapis ośmiowy

## Parameters

<i>calkowita_int</i>	Czesc calkowita wyniku wyrazenia
<i>ulamkowa</i>	Czesc ulamkowa wyniku wyrazenia
<i>precyzja</i>	Stala decydujaca o ilosci cyfr po przecinku

**2.1.1.4 konwersja\_na\_postfixs()**

```
vector< string > konwersja_na_postfixs (
```



```
vector< string > zapis_infiksowy,  
unordered_map< string, int > & operacja )
```

Funkcja zamieniająca zapis infiksowy wyrażenia na zapis postfiksowy.

Funkcja pobiera wektor z symbolami w zapisie infiksowym (domyslnym) i zmienia ich kolejność na postfiks (znak operacji występuje po liczbach), co umożliwia obliczanie wyrażen z wykorzystaniem stosu

#### Parameters

<i>zapis_infiksowy</i>	Wektor typu string, jako który podajemy wektor poddany funkcji <a href="#">rozbiecie()</a>
<i>operacja</i>	Mapa zawierająca operacje matematyczne

#### Returns

Wektor z pojedynczymi symbolami w kolejności postfiksowej

#### 2.1.1.5 oblicz\_wyrazenie()

```
float oblicz_wyrazenie (  
    vector< string > & zapis_postfiksowy,  
    unordered_map< string, int > & operacja,  
    string wartosci_pamieci[] )
```

Funkcja obliczająca wyrażenie z zapisu postfiksowego.

Funkcja pobiera wektor symboli w zapisie postfiksowym i oblicza właściwy wynik wykorzystując operacje stosu

#### Parameters

<i>zapis_postfiksowy</i>	Wektor typu string, jako który podajemy wektor poddany funkcji <a href="#">konwersja_na_postfiks()</a>
<i>operacja</i>	Mapa zawierająca operacje matematyczne
<i>wartosci_pamieci</i>	Tablica przechowująca wartości pamięci, z których funkcja odczytuje wartości

#### Returns

Wynik wyrażenia matematycznego

#### 2.1.1.6 rozbiecie()

```
vector< string > rozbiecie (  
    string & wyrazenie,  
    unordered_map< string, int > & operacja )
```

Funkcja rozbijająca podane wyrażenie na pojedyncze symbole.

Funkcja pobiera wyrażenie matematyczne podane przez użytkownika i rozбивa je na pojedyncze symbole w wektorze umożliwiając zastosowanie odwrotnej notacji polskiej przez funkcję [konwersja\\_na\\_postfiks\(\)](#).

**Parameters**

<i>wyrazenie</i>	Wyrażenie matematyczne wydobyte z wejścia
<i>operacja</i>	Mapa zawierająca operacje matematyczne

**Returns**

Wektor typu string z pojedynczymi symbolami (liczby lub znaki)

**2.1.1.7 wczytaj\_pamiec()**

```
void wczytaj_pamiec (
    string & nazwa_pliku,
    const int & N,
    string wartosci_pamieci[],
    bool & ui_plik )
```

Funkcja wczytująca wartości pamięci z pliku.

Funkcja otwiera plik o podanej nazwie i pobiera wartości pamięci z pliku do tablicy, umożliwiając ich odczyt

**Parameters**

<i>nazwa_pliku</i>	Nazwa pliku podana na wejściu
<i>N</i>	Stała określająca rozmiar tablicy wartości pamięci
<i>wartosci_pamieci</i>	Tablica przechowująca pobrane wartości pamięci
<i>ui_plik</i>	Zmienna zamykająca program przy niepowodzeniu otwarcia pliku

**2.1.1.8 zapisz\_pamiec()**

```
void zapisz_pamiec (
    string & nazwa_pliku,
    const int & N,
    string wartosci_pamieci[],
    bool & ui_plik )
```

Funkcja zapisująca wartości pamięci do pliku.

Funkcja otwiera plik o podanej nazwie i zapisuje do pliku wartości pamięci przechowywane w tablicy

**Parameters**

<i>nazwa_pliku</i>	Nazwa pliku podana na wejściu
<i>N</i>	Stała określająca rozmiar tablicy wartości pamięci
<i>wartosci_pamieci</i>	Tablica przechowująca wartości pamięci do zapisu
<i>ui_plik</i>	Zmienna zamykająca program przy niepowodzeniu otwarcia pliku

## 2.2 header.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002 using namespace std;
00010 vector<string>rozbicie(string& wyrazenie, unordered_map<string, int>& operacja);
00011
00019 vector<string>konwersja_na_postfiks(vector<string> zapis_infiksowy, unordered_map<string, int>&
    operacja);
00020
00028 float oblicz_wyrazenie(vector<string>& zapis_postfiksowy, unordered_map<string, int>& operacja, string
    wartosci_pamieci[]);
00029
00036 void dziesietna_bin(int& calkowita_int, float& ulamkowa, const int& precyzja);
00037
00044 void dziesietna_hex(int& calkowita_int, float& ulamkowa, const int& precyzja);
00045
00052 void dziesietna_oct(int& calkowita_int, float& ulamkowa, const int& precyzja);
00053
00061 void zapisz_pamiec(string& nazwa_pliku, const int& N, string wartosci_pamieci[], bool& ui_plik);
00062
00070 void wczytaj_pamiec(string& nazwa_pliku, const int& N, string wartosci_pamieci[], bool& ui_plik);
```

## 2.3 main.cpp File Reference

```
#include <iostream>
#include <cstdlib>
#include <vector>
#include <string>
#include <unordered_map>
#include <fstream>
#include "header.h"
```

## Functions

- int [main](#) (int argc, char \*argv[])
- void [wczytaj\\_pamiec](#) (string &nazwa\_pliku, const int &N, string wartosci\_pamieci[], bool &ui\_plik)  
*Funkcja wczytująca wartości pamięci z pliku.*
- void [zapisz\\_pamiec](#) (string &nazwa\_pliku, const int &N, string wartosci\_pamieci[], bool &ui\_plik)  
*Funkcja zapisująca wartości pamięci do pliku.*
- void [dziesietna\\_bin](#) (int &calkowita\_int, float &ulamkowa, const int &precyzja)  
*Funkcja konwertująca liczbę dziesiętną na binarną.*
- void [dziesietna\\_hex](#) (int &calkowita\_int, float &ulamkowa, const int &precyzja)  
*Funkcja konwertująca liczbę dziesiętną na szesnastkową.*
- void [dziesietna\\_oct](#) (int &calkowita\_int, float &ulamkowa, const int &precyzja)  
*Funkcja konwertująca liczbę dziesiętną na ośmiową.*
- float [oblicz\\_wyrazenie](#) (vector< string > &zapis\_postfiksowy, unordered\_map< string, int > &operacja, string wartosci\_pamieci[])  
*Funkcja obliczająca wyrażenie z zapisu postfiksowego.*
- vector< string > [konwersja\\_na\\_postfiks](#) (vector< string > zapis\_infiksowy, unordered\_map< string, int > &operacja)  
*Funkcja zamieniająca zapis infiksowy wyrażenia na zapis postfiksowy.*
- vector< string > [rozbicie](#) (string &wyrazenie, unordered\_map< string, int > &operacja)  
*Funkcja rozbijająca podane wyrażenie na pojedyncze symbole.*

## 2.3.1 Function Documentation

### 2.3.1.1 dziesiętna\_bin()

```
void dziesiętna_bin (
    int & całkowita_int,
    float & ułamkowa,
    const int & precyzja )
```

Funkcja konwertująca liczbę dziesiętną na binarną.

Funkcja pobiera osobno część całkowitą i ułamkową wyniku, a następnie zamienia je w zapis binarny

#### Parameters

<i>całkowita_int</i>	Część całkowita wyniku wyrażenia
<i>ułamkowa</i>	Część ułamkowa wyniku wyrażenia
<i>precyzja</i>	Stała decydująca o ilości cyfr po przecinku

### 2.3.1.2 dziesiętna\_hex()

```
void dziesiętna_hex (
    int & całkowita_int,
    float & ułamkowa,
    const int & precyzja )
```

Funkcja konwertująca liczbę dziesiętną na szesnastkową.

Funkcja pobiera osobno część całkowitą i ułamkową wyniku, a następnie zamienia je w zapis szesnastkowy

#### Parameters

<i>całkowita_int</i>	Część całkowita wyniku wyrażenia
<i>ułamkowa</i>	Część ułamkowa wyniku wyrażenia
<i>precyzja</i>	Stała decydująca o ilości cyfr po przecinku

### 2.3.1.3 dziesiętna\_oct()

```
void dziesiętna_oct (
    int & całkowita_int,
    float & ułamkowa,
    const int & precyzja )
```

Funkcja konwertująca liczbę dziesiętną na ośmiową.

Funkcja pobiera osobno część całkowitą i ułamkową wyniku, a następnie zamienia je w zapis ośmiowy

#### Parameters

<i>calkowita_int</i>	Część całkowita wyniku wyrażenia
<i>ulamkowa</i>	Część ułamkowa wyniku wyrażenia
<i>precyzja</i>	Stała decydująca o ilości cyfr po przecinku

#### 2.3.1.4 konwersja\_na\_postfiks()

```
vector< string > konwersja_na_postfiks (
    vector< string > zapis_infiksowy,
    unordered_map< string, int > & operacja )
```

Funkcja zamieniająca zapis infiksowy wyrażenia na zapis postfiksowy.

Funkcja pobiera wektor z symbolami w zapisie infiksowym (domyslnym) i zmienia ich kolejność na postfiks (znak operacji występuje po liczbach), co umożliwia obliczanie wyrażen z wykorzystaniem stosu

#### Parameters

<i>zapis_infiksowy</i>	Wektor typu string, jako który podajemy wektor poddany funkcji <a href="#">rozbić()</a>
<i>operacja</i>	Mapa zawierająca operacje matematyczne

#### Returns

Wektor z pojedynczymi symbolami w kolejności postfiksowej

#### 2.3.1.5 main()

```
int main (
    int argc,
    char * argv[] )
```

#### 2.3.1.6 oblicz\_wyrazenie()

```
float oblicz_wyrazenie (
    vector< string > & zapis_postfiksowy,
    unordered_map< string, int > & operacja,
    string wartosci_pamieci[] )
```

Funkcja obliczająca wyrażenie z zapisu postfiksowego.

Funkcja pobiera wektor symboli w zapisie postfiksowym i oblicza właściwy wynik wykorzystując operacje stosu

## Parameters

<i>zapis_postfiksowy</i>	Wektor typu string, jako który podajemy wektor poddany funkcji <a href="#">konwersja_na_postfiks()</a>
<i>operacja</i>	Mapa zawierająca operacje matematyczne
<i>wartosci_pamieci</i>	Tablica przechowująca wartości pamięci, z których funkcja odczytuje wartości

## Returns

Wynik wyrażenia matematycznego

**2.3.1.7 rozbiecie()**

```
vector< string > rozbiecie (
    string & wyrazenie,
    unordered_map< string, int > & operacja )
```

Funkcja rozbijająca podane wyrażenie na pojedyncze symbole.

Funkcja pobiera wyrażenie matematyczne podane przez użytkownika i rozбивa je na pojedyncze symbole w wektorze umożliwiając zastosowanie odwrotnej notacji polskiej przez funkcję [konwersja\\_na\\_postfiks\(\)](#).

## Parameters

<i>wyrazenie</i>	Wyrażenie matematyczne wydobyte z wejścia
<i>operacja</i>	Mapa zawierająca operacje matematyczne

## Returns

Wektor typu string z pojedynczymi symbolami (liczby lub znaki)

**2.3.1.8 wczytaj\_pamiec()**

```
void wczytaj_pamiec (
    string & nazwa_pliku,
    const int & N,
    string wartosci_pamieci[],
    bool & ui_plik )
```

Funkcja wczytująca wartości pamięci z pliku.

Funkcja otwiera plik o podanej nazwie i pobiera wartości pamięci z pliku do tablicy, umożliwiając ich odczyt

## Parameters

<i>nazwa_pliku</i>	Nazwa pliku podana na wejściu
<i>N</i>	Stała określająca rozmiar tablicy wartości pamięci
<i>wartosci_pamieci</i>	Tablica przechowująca pobrane wartości pamięci
<i>ui_plik</i>	Zmienna zamykająca program przy niepowodzeniu otwarcia pliku

### 2.3.1.9 zapisz\_pamiec()

```
void zapisz_pamiec (
    string & nazwa_pliku,
    const int & N,
    string wartosci_pamieci[],
    bool & ui_plik )
```

Funkcja zapisująca wartości pamięci do pliku.

Funkcja otwiera plik o podanej nazwie i zapisuje do pliku wartości pamięci przechowywane w tablicy

#### Parameters

<i>nazwa_pliku</i>	Nazwa pliku podana na wejściu
<i>N</i>	Stała określająca rozmiar tablicy wartości pamięci
<i>wartosci_pamieci</i>	Tablica przechowująca wartości pamięci do zapisu
<i>ui_plik</i>	Zmienna zamykająca program przy niepowodzeniu otwarcia pliku





# Index

- dziesietna\_bin
  - header.h, [3](#)
  - main.cpp, [8](#)
- dziesietna\_hex
  - header.h, [4](#)
  - main.cpp, [8](#)
- dziesietna\_oct
  - header.h, [4](#)
  - main.cpp, [8](#)
- header.h, [3](#)
  - dziesietna\_bin, [3](#)
  - dziesietna\_hex, [4](#)
  - dziesietna\_oct, [4](#)
  - konwersja\_na\_postfiks, [4](#)
  - oblicz\_wyrazenie, [5](#)
  - rozbicie, [5](#)
  - wczytaj\_pamiec, [6](#)
  - zapisz\_pamiec, [6](#)
- konwersja\_na\_postfiks
  - header.h, [4](#)
  - main.cpp, [9](#)
- main
  - main.cpp, [9](#)
- main.cpp, [7](#)
  - dziesietna\_bin, [8](#)
  - dziesietna\_hex, [8](#)
  - dziesietna\_oct, [8](#)
  - konwersja\_na\_postfiks, [9](#)
  - main, [9](#)
  - oblicz\_wyrazenie, [9](#)
  - rozbicie, [10](#)
  - wczytaj\_pamiec, [10](#)
  - zapisz\_pamiec, [11](#)
- oblicz\_wyrazenie
  - header.h, [5](#)
  - main.cpp, [9](#)
- rozbicie
  - header.h, [5](#)
  - main.cpp, [10](#)
- wczytaj\_pamiec
  - header.h, [6](#)
  - main.cpp, [10](#)
- zapisz\_pamiec
  - header.h, [6](#)
  - main.cpp, [11](#)