AiSD Laboratorium 1 LIFEBUOY

W tym pliku zawarte są częściowe rozwiązania (raczej wskazówki) do zadań z laboratorium 1.

Wskazówka do zadania 2.

Metoda brutalna

```
/*
 * AiSD lab 1
 * Zadanie 2.
 * Rozwiązanie bezpośrednie
 */

#include <fstream>

int main(){

std::ofstream out("size_of_buildin_types.txt");

    out << "char: " << sizeof(char) << '\n';

    ...

out.close();
}
```

Zestosowanie makro definicji

```
/*
 * AiSD lab 1
 * Zadanie 2.
 * Rozwiązanie z użyciem makro definicji
 */

#include <fstream>

// MSG(char) rozwija się do << "char: " << sizeof(char) << '\n'
// pamiętać o dodaniu ; po
#define MSG(arg) << #arg ": " << sizeof(arg) << '\n'

int main(){

std::ofstream out("size_of_buildin_types.txt");

    out << "Rozmiary typów wbudowanych:\n"
        MSG(char)
        MSG(unsigned char)
    ...

out.close();
}
```

Zastosowanie szablonów

```
/*
 * AiSD lab 1
 * Zadanie 2.
 * Rozwiązanie z użyciem szablonu
 */

#include <fstream>
#include <string>

template<class T>
std::string line(const std::string& type_name){

    return type_name + ": " + std::to_string(sizeof(T)) + "\n";
}

int main(){

std::ofstream out("size_of_buildin_types.txt");

    out << "Rozmiary typów wbudowanych:\n"
        << line<char>("char")
        << line<unsigned char>("unsigned char")
    ...

    out.close();
}
```

Modyfikacja powyższego

Zapis do pliku następuje dopiero po wyprodukowaniu całości

```
/*
 * AiSD lab 1
 * Zadanie 2.
 * Rozwiązanie z użyciem szablonu i pomocnicznych zmiennych
 */

#include <fstream>
#include <string>
#include <sstring>

template<class T>
std::string line(const std::string& type_name){

    return type_name + ": " + std::to_string(sizeof(T)) + "\n";
}


int main(){

std::ostringstream out_stringstream {};

    out_stringstream << line<char>("char")
                     << line<int>("int")
    ...
```

```
std::string out_string {"Rozmiary typów wbudowanych:\n"};

    out_string += line<char>("char")
                + line<unsigned char>("unsigned char")
    ...


// otwiera plik do zapisu,
// jeżeli plik nie istnieje to go tworzy
// jeżeli istnieje to nadpisuje zawartość!
std::ofstream out("size_of_buildin_types.txt");

    out << out_string;

// std::stringstream::str() zwraca obiekt std::string
    out << out_stringstream.str();

    out.close();
}
```

Wskazówki do zadania 3.

- rozwiązanie bezpośrednie

```
out << " " << std::numeric_limits<int>::lowest()
    << " " << std::numeric_limits<int>::min()
    << " " << std::numeric_limits<int>::max()
    << " " << std::numeric_limits<int>::digits10
...
```

- z użyciem makra

```
#define MSG(arg) << " " << std::numeric_limits<int>::lowest()\
                 << " " << std::numeric_limits<int>::min()
...
out MSG(int)
    MSG(float)
...
```

- z użyciem szablonów

```
template<typename T>
std::string prepare_line(std::string name){

    out << " " << std::numeric_limits<T>::lowest()
...
}
```

- clasa `string` posiada konstruktor postaci `string(k,ch)`, który inicjalizuje obiekt za pomocą k kopii znaku `ch` (lub funkcja `assign(i,ch)`)