

## Zadanie F: Program do wszystkiego (20 punktów)

Najnowszy projekt Awesome Studios Inc. to program do robienia wszystkiego. Wszystko jest osiągnięte poprzez (prawie) dowolne pluginy które mogą zostać napisane i dodane do programu.

W chwili obecnej Awesome Studios Inc. pracuje nad wersją prototypową i Tobie przypadło napisanie takiego programu.

Program czeka na polecenia podawane w standardowym wejściu. Każde polecenie znajduje się w osobnej linijce. Polecenie składa się z nazwy komendy, oraz listy argumentów oddzielonych pojedynczymi spacjami. Początkowo Twój program powinien rozpoznawać dwie komendy:

- **end** – kończy działanie programu
- **load <plik>** – wczytuje podaną bibliotekę dynamiczną

Jeśli wczytanie biblioteki powiedzie się, program powinien wywołać z niej funkcję:

```
char** info()
```

która informuje program główny o wszystkich funkcjach dostępnych w pluginie. Każda funkcja pluginu zapisana jest na dwóch pozycjach w tablicy:

```
<nazwa> <typ>
```

gdzie **<nazwa>** to dowolna, dozwolona w C nazwa funkcji, zaś **<typ>** to jedna z wartości **int**, **float**, **int[]**, lub **float[]** określająca typ argumentu funkcji.

Funkcje określone jako **int** i **float** oczekują jednego argumentu w tym typie. Funkcje określone jako **int[]** i **float[]** oczekują dwóch argumentów: pierwszy, typu **int** określa wielkość tablicy, a drugi to wskaźnik na ową tablicę (odpowiednio **int\*** lub **float\***). Wszystkie funkcje zwracają **void**.

Lista funkcji zakończona jest symbolem **NULL** (t.j. wskaźnik **char\*** ma wartość **NULL**) Wszystkie funkcje załadowane z biblioteki powinny stać się dostępne dla użytkownika i wywoływane są za pomocą nazwy i argumentu/argumentów, oddzielonych spacją.

- Jeśli plik biblioteki nie istnieje, program powinien napisać **File not found!** i przejść do następnej linijki.
- Jeśli dana funkcja nie istnieje, program powinien napisać **Function not found!** i przejść do następnej linijki. Funkcje które nie istnieją nie oczekują żadnych argumentów.
- Poza powyższymi przypadkami można założyć że polecenia podawane są poprawnie. Cały program zawsze kończy się **end**.
- Dopuszczalne nazwy funkcji występujące w bibliotekach są takie jak w języku C.
- Jeśli biblioteka deklaruje funkcję której nazwa wcześniej już się pojawiała, to nowa wersja nadpisuje starą.
- Można założyć, że całkowita liczba funkcji nie przekroczy 256.
- Tablice zwracanej przez **info()** nie należy usuwać.

Akceptowane są pliki źródłowe języka C lub C++.

- Dla języka C kompilujemy za pomocą **gcc -std=c11 <plik>.c -ldl -Wl,-rpath=. -o <plik>**

- Dla języka C++ kompilujemy za pomocą `g++ -std=c++14 -fpermissive <plik>.cpp -ldl -Wl,-rpath=. -o <plik>`

warningi podczas kompilacji są ignorowane i nie wpływają na poprawność rozwiązania.

### Przykład:

Założmy, że mamy bibliotekę `libhello.so` skompilowaną ze źródła `hello.c`:

```
#include "stdio.h"

const char** info() {
    static const char* info[] = { "hello", "int", NULL };
    return info;
}

void hello(int c) {
    for (int i=0; i<c; ++i)
        printf("Hello world!\n");
}
```

Plik wejściowy:

```
hello
load libhello.so
hello 2
end
```

Wygeneruje wynik na standardowym wyjściu:

```
Function not found!
Hello world!
Hello world!
```