

6.14 Binarna grafika żółwiowa – wersja 4

Zadaniem jest napisanie programu, który dokonuje translacji binarnie zakodowanych komend sterujących żółwiem na rastrowy plik w formacie BMP [5, 16]. Program ma składać się z części napisanej w języku C/C++, z której wywoływana jest funkcja napisana w assemblerze procesora Intel x86 (32 bitowego). Pisząc program należy zaimplementować funkcję assemblerową o następującej deklaracji w języku C:

```
int exec_turtle_cmd(unsigned char *dest_bitmap,
                   unsigned char *command,
                   TurtleContextStruct *tc)
```

Opis:

Funkcja wykonuje jedną komendę grafiki żółwiowej znajdującą się w buforze komend.

Specyfikacja komend jest taka sama jak w zadaniu 3.14.

Wartość zwracana:

- 0 - w przypadku powodzenia
- liczba większa od 0 – numer błędu w przypadku, gdy wykonanie komendy nie powiodło się.

Parametry funkcji:

- **dest_bitmap** - wskazanie na bufor zawierający obraz (łącznie z nagłówkiem BMP),
- **command** – wskazanie na bufor zawierający binarną komendę grafiki żółwiowej,
- **tc** – struktura przechowująca kontekst żółwia - służy do przekazania dodatkowych danych potrzebnych do prawidłowego wykonania sekwencji komend np.:

```
struct TurtleContextStruct
{
    int x_pos; //współrzędna x położenia żółwia
    int y_pos; //współrzędna y położenia żółwia
               //i ewentualnie inne potrzebne pola
}
```

Wejście

- Plik binarny zawierający 16/32-bitowe komendy sterujące żółwiem
- Nazwa pliku: "input.bin"
- Z pliku tekstowego "config.txt" należy pobrać dwie liczby całkowite (każda w oddzielnym wierszu) oznaczające rozmiar obrazu w pikselach.

Wyjście

- Plik BMP zawierający wygenerowany obraz:
 - Sub format: 24 bitowy RGB – bez kompresji,
 - Rozmiar obrazu: dowolny,
 - Nazwa pliku: "output.bmp"

Uwagi

1. Odczyt danych z pliku realizowany jest na poziomie języka C.
2. Alokacja pamięci na bufory realizowana jest na poziomie języka C.
3. Nagłówek pliku BMP można wygenerować na poziomie języka C.
4. Nie wolno interpretować ani przetwarzać komend żółwia w języku C – jest to zadanie funkcji asemblerowej.
5. Proszę zwrócić uwagę na możliwość wystąpienia bajtów dopełniających (ang. padding bytes) w przypadku, gdy liczba bajtów w wierszu obrazu nie jest podzielna przez 4.
6. Z projektem należy dostarczyć plik `makefile`, przy pomocy którego zostanie automatycznie wygenerowany plik wykonywalny o nazwie **`turtle`**.
7. Projekt należy dostarczyć w formie jednego pliku **`.zip`** lub **`.tar.gz`**.