Dokumentacja projektu „3.10 Kod kreskowy - Code 39”

# Struktura danych i implementacja:

1. **Code**

Code .space 60: Miejsce w pamięci na zdekodowany kod kreskowy (max 60 znaków).

1. **Znak**

Znak: .asciiz "000000000\n": String, do którego zapisuję konfiguracje szerokości kresek i przerw (1 dla wąskiej, 2 dla szerokiej). Służy do rozpoznania zakodowanego znaku kodu.   
„\n” na końcu użyty dla przejrzystości testów.

1. **Tablica\_znakow**

Tablica\_znakow: .asciiz "\*0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ-. $/+%" : Znaki wypisane w tej samej kolejności co odpowiadające im konfiguracje szerokości kresek i przerw.

1. **Tablica\_kresek**

Tablica\_kresek: .asciiz "121121211.111221211.211211112.112211112….” : Wszystkie konfiguracje znaków.

# Implementacja algorytmu:

1. Wczytanie 54 bajtów nagłówka, sprawdzenie wymiarów
2. Wczytanie obrazka
3. Przesunięcie wskaźnika na środek obrazka (600px\*3bajty\*50/2px= 45000), następnie dopóki nie napotka czarnej składowej piksela (skoki co 3 bajt, bo zakładam, że paski i tło będą w kolorach czarny i biały)
4. Zapisanie szerokości pasków w rejestrze t1 (liczba pikseli wąskiego), t2 (szerokiego)
5. Wyliczenie liczby bajtów pozostałych pasków znaku start (5\*t1\*3+2\*t2\*3) i przesunięcie wskaźnika za znak start
6. Zapisuję konfigurację kresek i przerw do tablicy Znak :



1. Dekodowanie znaku

Ustawiam wskaźnik na początek tablicy Znak, idę po kolejnych znakach tablic Znak i Tablica\_kresek. Wczytuję po 1 bajcie. Jeśli trafię na kropkę w Tablica\_kresek- znak znaleziony, kończę pętlę. Jeśli nie, przesuwam wskaźniki o 1 na obu tablicach. Sprawdzam czy są różne- jeśli tak, przesuwam wskaźnik na Tablica\_kresek za kropkę, a wskaźnik na Tablica\_znakow na kolejny znak. Zaczynam porównywanie od początku. Jeśli znak znaleziony, sprawdzam czy nie jest to \*. Jeśli tak- kończę wczytywanie. Jeśli nie: inkrementuję licznik znaków w kodzie, zapisuję znak w tablicy Code i przesuwam na niej wskaźnik o 1 powtarzam algorytm.

1. Liczę znak kontrolny

Cofam wskaźnik na tablicy Code i zapisuję w rejestrze ostatni znak kodu- powinien być sumą modulo znaków kodu. Zeruję rejestr, który będzie licznikiem pętli. Ustawiam wskaźnik na początek Code. Pętla: inkrementuję licznik pętli i sprawdzam czy nie jest równa liczbie znaków. Jeśli kod będzie zawierał 1 znak, od razu wyjdę z pętli, porównam jedyny znak w kodzie z wyzerowanym rejestrem i zwrócę komunikat o błędzie. Kontynuacja: wczytuję do rejestru bajt z Code , przesuwam wskaźnik na Code na kolejny znak, ustawiam wskaźnik na tablicę\_znaków, zeruję licznik pozycji w tej tablicy i wchodzę w pętlę zagnieżdżoną. Wyszukuję znak w tablicy, inkrementując licznik. Gdy znajdę już pozycję znaku, dodaję tę liczbę do rejestru i sprawdzam czy jego wartość wykracza poza 42. Jeśli tak, odejmuję od sumy 43. Powtarzam algorytm dopóki licznik pętli nie będzie równy liczbie znaków w kodzie.

1. Dekoduję znak kontrolny

Na tablicy Tablica\_znaków przesuwam wskaźnik o 1, bo \* jest u mnie na początku a nie na końcu. Następnie przesuwam go o wyliczoną sumę kontrolną i sprawdzam, czy ten znak jest równy ostatniemu w tablicy. Jeśli nie- komunikat o błędzie i wyjście z programu.

1. Wyświetlam kod

Na ostatniej pozycji tablicy Code zastępuję ostatni znak znakiem nowej linii. Wyświetlam kod. Zamykam plik.

# Testy:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kod | Szerokość wąskiego paska [piksele] | Znak kontrolny | Wynik testu | Powodzenie testu |
| 548SD548SFD | 1 | 2 - obecny | Kod | Pomyślny |
| %%%% | 1 | $ - obecny | Kod | Pomyślny |
| 548SD%548%SFD | 1 | 0 - obecny | Kod | Pomyślny |
| 548SD548SFD | 1 | 2 - brak | Komunikat o błędzie | Pomyślny |
| 1 | 1 | 1 - brak | Komunikat o błędzie | Pomyślny |
| 11 | 1 | 1 - obecny | Kod | Pomyślny |
| 316+-%1 | 2 | 1 - obecny | Kod | Pomyślny |
| 1876214598536 | 2 | M - obecny | Kod | Pomyślny |
| % | 2 | % - obecny | Kod | Pomyślny |
| 1 | 2 | 1 - brak | Komunikat o błędzie | Pomyślny |
| 134 | 3 | A - obecny | Kod | Pomyślny |
| 316+-%1 | 3 | 1 - obecny | Kod | Pomyślny |
| %%%% | 3 | $ - obecny | Kod | Pomyślny |
| 73+ | 3 | 8 - brak | Komunikat o błędzie | Pomyślny |
| 316+-%1 | 4 | 1 - obecny | Kod | Pomyślny |
| 12 | 7 | 3 - obecny | Kod | Pomyślny |
| Brak obrazka pod ścieżką | - | - | Komunikat o błędzie | Pomyślny |
| Niepoprawny rozmiar | - | - | Komunikat o błędzie | Pomyślny |
| Niepoprawny format | - | - | Komunikat o błędzie | Pomyślny |

Program w żadnym testowanym przypadku nie wszedł w nieskończoną pętlę ani nie wypisał niepoprawnego kodu.