Obraz zawierający tekst, Czcionka, logo, Grafika

Opis wygenerowany automatycznie

Sprawozdanie 2

Ćwiczenie 2. KODY KRESKOWE (EAN)

Autorzy: Krzysztof Zalewa, Wiktor Wojnar

Spis treści

[1.Wstęp 2](#_Toc181718819)

[1.1 Kody EAN 2](#_Toc181718820)

[1.1.1 Organizacja zarządzająca 2](#_Toc181718821)

[1.1.2 Struktura administracyjna 3](#_Toc181718822)

[1.1.3 Procedury uzyskiwania 3](#_Toc181718823)

[1.3 Kody EAN13 4](#_Toc181718824)

[1.3.1 Budowa 4](#_Toc181718825)

[1.3.2 Alfabety 5](#_Toc181718826)

[1.3.4 Samosprawdzalność 6](#_Toc181718827)

[1.2.4 Wymiary kodu 7](#_Toc181718828)

[2.Zadanie Laboratoryjne 7](#_Toc181718829)

[2.1 Treść zadania 7](#_Toc181718830)

[2.2 Opis działania programu 7](#_Toc181718831)

[2.3 Kod programu 7](#_Toc181718832)

[3.Wnioski 12](#_Toc181718833)

[4. Źródła 12](#_Toc181718834)

# 1.Wstęp

## 1.1 Kody EAN

### 1.1.1 Organizacja zarządzająca

Organizacją zarządzającą przyznawaniem kodów EAN jest GS1. GS1 jest organizacją non-profit zajmującą się opracowaniem i utrzymywaniem standardów w biznesie. Obecnie GS1 zrzesza 115 lokalnych Organizacji członkowskich oraz ponad 2 miliony zarejestrowanych firm.

### 1.1.2 Struktura administracyjna

Obraz zawierający tekst, zrzut ekranu, Czcionka, Równolegle

Opis wygenerowany automatycznie

Rysunek 1. Schemat struktury GS1[1]

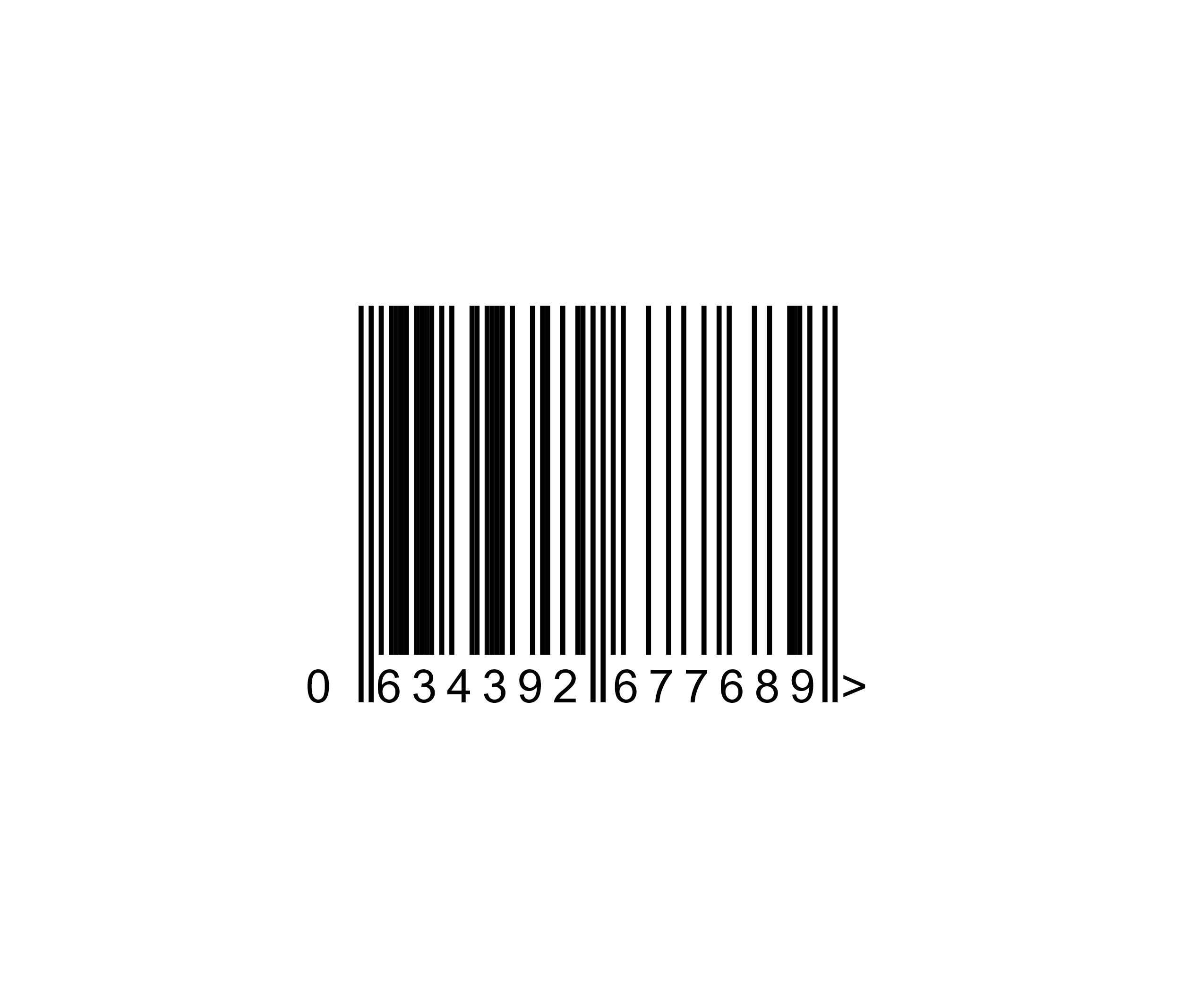
Organizacja GS1 składa się z 3 głównych poziomów :

1. Zgromadzenie Ogólne (ang. General Assembly) – składa się z przedstawicieli wszystkich organizacji członkowskich
2. Rada zarządzająca (ang Management Board) i lokalne zarządy (Local GS1 Boards) – rada zarządzająca składa się z przedstawicieli największych organizacji członkowskich (np. GS1 Chiny, Stany zjednoczone, Niemcy) i koncernów (np. Google, Dr Oetker, L’Oréal) [2]. Te isnstytucje odpowiadają za wdrażanie globalnych kierunków strategicznych.
3. Globalne biuro (ang Global Office) przewodzące i koordynujące prace nad opracowywaniem i utrzymywaniem nowych standardów. Oraz organizacje członkowskie (ang. Local GS1 Memeber Organisation) odpowiedzialne za wdrażanie lokalnych strategii/

### 1.1.3 Procedury uzyskiwania

Pozyskanie kodu EAN jest stosunkowo proste. Należy wejść na stronę GS1 (Nie musi być polska, po wybraniu lokalizacji w Polsce nastąpi przekierowanie na stronę polskiego GS1). Następnie należy podać szacowaną ilość produktów i dane firmy dla której tworzymy kod. Na koniec należy dokonać opłaty rejestracyjnej. Kod przydzielony zostanie na zasadzie subskrypcji więc występować będzie opłata roczna.[3]

## 1.3 Kody EAN13



Rysunek 2. Przykładowy kod EAN13 wygenerowany przy pomocy naszej aplikacji

### 1.3.1 Budowa

Kod EAN13 składa się z sekwencji kresek i przerw, pod kodem kreskowym znajduje się ten sam kod w formie która czytelna jest dla człowieka zakończony znakiem ‘>’. Kod liczbowy składa się z 12 liczb podzielonych na 4 obszary.

1. Pierwsze obszar (pierwsze 3 liczby) to numer organizacji krajowej która przyznała dany kod
2. Drugi obszar (kolejne 4 -7) liczb to kod identyfikujący wytwórcę produktu
3. Trzeci obszar (jego długość jest zależna od obszaru 2) to kod identyfikujący produkt
4. Ostatni obszar to cyfra kontrolna

Kody identyfikujące agencje są zawsze 3 cyfrowe, jednakże czasami zapisywane są tylko dwie liczby. Oznacza to że w danym państwie jest wiele agencji odpowiadających za przyznawanie kodów EAN13.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

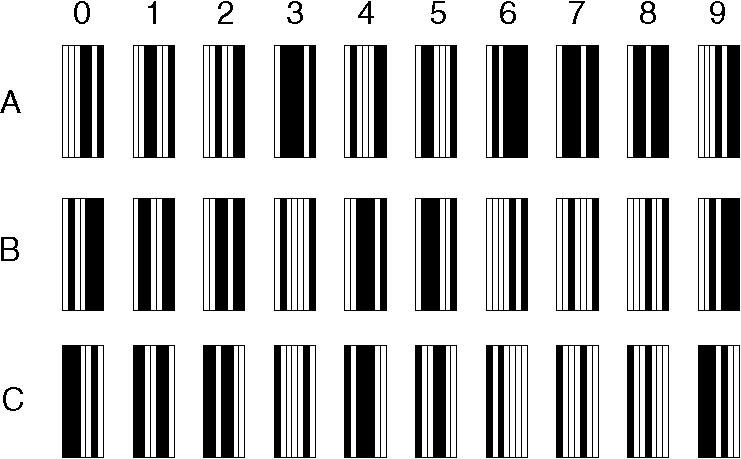
Rysunek 3. Wybrane kody identyfikujące agencje [8]

Na przykład w Chorwacji istniej organizacja GS1 której numer to 385. Natomiast w Federacji Rosyjskiej lokalnej organizacji przyznano numery 460 – 469. ( W Niemczech numery lokalnej organizacji to 400 – 449).

Długość obszarów 2 i 3 jest różna w zależności od potrzeb wytwórcy (Firmy o niewielkiej liczbie produktów nie potrzebują 5! różnych kodów).

### 1.3.2 Alfabety

W kodzie EAN13 każda cyfra reprezentowana jest poprzez 7 bitową sekwencję kresek i przerw. Dla każdej liczby są trzy takie sekwencje. Dlatego też w kodach EAN13 są trzy alfabety. Zazwyczaj oznaczane jako A,B i C (Rysunek 4). Sekwencje te zawsze składają się z dwóch pasków białych i dwóch czarnych . Paski te mogą mieć szerokość od jednej do czterech kresek.



Rysunek 4. Wizualizacja trzech alfabetów [9]

### 1.3.4 Samosprawdzalność

Sekwencje w alfabetach kodu EAN13. Są nietrywialnie różne od siebie, powoduje to że nawet w przypadku błędu w druku trudno jest popełnić błąd w odczycie. By umożliwić poprawne skanowanie odwróconego kodu używa się mieszanki alfabetów A,B i C. Pierwsza liczba w zakodowana jest przy pomocy zestawu znaków użytego do zakodowania kolejnych sześciu cyfr (Tabela 1 ). Następne sześć liczb zakodowane jest mieszanką alfabetów A i B (Rysunek 5 W naszym kodzie pierwsza liczba to 0 więc wszystkie pozostałe kodowane są w alfabecie A). Ostatnie sześć liczb kodowane jest przy pomocy alfabetu C. Dzięki temu urządzenie skanujące może wykryć że czyta kod od końca (kod jest odwrócony) i przekształcić go tak by został odczytany poprawnie

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Pierwszy znak** | **Zbiór według którego są kodowane kolejne znaki** | | | | | | | | | | | |
| **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
| 0 | A | A | A | A | A | A | C | C | C | C | C | C |
| 1 | A | A | B | A | B | B | C | C | C | C | C | C |
| 2 | A | A | B | B | A | B | C | C | C | C | C | C |
| 3 | A | A | B | B | B | A | C | C | C | C | C | C |
| 4 | A | B | A | A | B | B | C | C | C | C | C | C |
| 5 | A | B | B | A | A | B | C | C | C | C | C | C |
| 6 | A | B | B | B | A | A | C | C | C | C | C | C |
| 7 | A | B | A | B | A | B | C | C | C | C | C | C |
| 8 | A | B | A | B | B | A | C | C | C | C | C | C |
| 9 | A | B | B | A | B | A | C | C | C | C | C | C |

Tabela 1.Sposób kodowania pierwszego znaku w EAN-13

Obraz zawierający tekst, zrzut ekranu, typografia, design

Opis wygenerowany automatycznie

Rysunek 5. Fragment wygenerowanego kodu EAN13

### 1.2.4 Wymiary kodu

Obraz zawierający tekst, linia, Prostokąt, typografia

Opis wygenerowany automatycznie

Rysunek 6. Wymiary kodów EAN [10]

# 2.Zadanie Laboratoryjne

## 2.1 Treść zadania

1. Opisać budowę kodu EAN13 oraz kodu QR,
2. Napisać program generujący (dla wprowadzanej sekwencji cyfr) kod EAN13.

## 2.2 Opis działania programu

Program pobiera od użytkownika numeryczny kod EAN do wygenerowania. Dana jest także możliwość zmiany rozmiaru generowanego kodu. Następnie upewnia się czy kod jest poprawny (13 cyfr, bez liter i znaków specjalnych). Kod jest generowany i zostaje zapisany w plikach output.svg (Wersja bez kodu numerycznego i widocznych znaków początku, środka i końca) i output2.svg. Wygenerowany kod ma rozmiary odpowiadające standardom.

## 2.3 Kod programu

//

// Created by Wiktor on 17.10.2024.

//

#include <cstdint>

#include <cstdio>

#include <string.h>

int32\_t EAN13\_WIDTH = 1 ,EAN13\_HEIGHT = 69;

double EAN13\_WIDTH\_MAX = 37.29, EAN13\_SCALE = 10.0, EAN13\_WIDTH\_D = 31.35/95.0, EAN13\_HEIGHT\_D = 22.85, EAN13\_MAX\_HEIGHT = 25.95, EAN13\_FIRST\_SPACE = 3.63, EAN13\_TEXT\_SIZE\_D = 3;

struct ean13{

    uint8\_t A[10] = {0b0001101, 0b0011001, 0b0010011,0b0111101, 0b0100011,

                     0b0110001,0b0101111,0b0111011,0b0110111,0b0001011};

    uint8\_t B[10] = {0b0100111, 0b0110011, 0b0011011,0b0100001, 0b0011101,

                     0b0111001,0b0000101,0b0010001,0b0001001,0b0010111};

    uint8\_t oddity[10] = {

        0b111111, 0b110100,0b110010,0b110001,0b101100, 0b100110, 0b100011, 0b101010, 0b101001, 0b100101

    };

};

///SVG

void initSVG(FILE \*svg, int width, int height){

    fprintf(svg , "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" );

    fprintf(svg , "<!DOCTYPE svg PUBLIC \"-//W3C//DTD SVG 1.1//EN\" " );

    fprintf( svg, "\"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd\">\n" );

    fprintf(svg , "<svg xmlns=\"http://www.w3.org/2000/svg\" width=\"%d\" height=\"%d\" viewBox=\"0 0 %d %d\" style=\"background-color: white\">\n", width, height, width, height );

    fprintf(svg, "<rect x=\"0\" y=\"0\" width=\"%d\" height=\"%d\" style=\"fill: white\"/>\n", width, height);

}

void initSVG\_d(FILE \*svg, double width, double height){

    fprintf(svg , "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" );

    fprintf(svg , "<!DOCTYPE svg PUBLIC \"-//W3C//DTD SVG 1.1//EN\" " );

    fprintf( svg, "\"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd\">\n" );

    fprintf(svg , "<svg xmlns=\"http://www.w3.org/2000/svg\" width=\"%f\" height=\"%f\" viewBox=\"0 0 %f %f\" style=\"background-color: white\">\n", width, height, width, height );

    fprintf(svg, "<rect x=\"0\" y=\"0\" width=\"%f\" height=\"%f\" style=\"fill: white\"/>\n", width, height);

}

void endSVG(FILE \*svg){

    fprintf( svg, "</svg>" );

    fclose( svg );

    //free( svg );

}

void svg\_line( FILE \*svg, int x1, int y1, int x2, int y2, int stroke\_width, unsigned int color )

{

    fprintf( svg, "<line x1=\"%d\" y1=\"%d\" x2=\"%d\" y2=\"%d\" style=\"stroke-width:%d; stroke:#%06x\" />\n", x1, y1, x2, y2, stroke\_width, color );

}

void svg\_line\_d( FILE \*svg, double x1, double y1, double x2, double y2, double stroke\_width, unsigned int color )

{

    fprintf( svg, "<line x1=\"%f\" y1=\"%f\" x2=\"%f\" y2=\"%f\" style=\"stroke-width:%f; stroke:#%06x\" />\n", x1, y1, x2, y2, stroke\_width, color );

}

void svg\_text\_d(FILE \*svg, double x1, double y1, double size, unsigned int color, const char text)

{

    fprintf(svg, "<text x=\"%f\" y=\"%f\" font-size=\"%f\" fill=\"#%06x\">%c</text>\n",

            x1, y1, size, color, text);

}

void encodeEan(const char\* code, char \* finalCode){

    struct ean13 \_ean;

    FILE \*file = fopen("output2.svg","w");

    /// Input validation

    for(int i = 0; i < 13; i++) {

        if (code[i] < '0' || code[i] > '9') {

            printf("Invalid key entered\n");

            return;

        }

    }

    int marginX = 200;

    int marginY = 200;

    double EAN13\_WIDTH\_D\_LOCAL = EAN13\_WIDTH\_D\*EAN13\_SCALE,

    EAN13\_HEIGHT\_D\_LOCAL = EAN13\_HEIGHT\_D\*EAN13\_SCALE+marginY,

    EAN13\_MAX\_HEIGHT\_LOCAL = EAN13\_MAX\_HEIGHT\*EAN13\_SCALE+marginY,

    EAN13\_FIRST\_SPACE\_LOCAL = EAN13\_FIRST\_SPACE\*EAN13\_SCALE,

    EAN13\_TEXT\_SIZE\_D\_LOCAL = EAN13\_TEXT\_SIZE\_D\*EAN13\_SCALE,

    EAN13\_WIDTH\_MAX\_LOCAL = EAN13\_WIDTH\_MAX\*EAN13\_SCALE+marginX;

    double loc= marginX;

    double width =  EAN13\_WIDTH\_MAX\_LOCAL + marginX;

    double height = EAN13\_MAX\_HEIGHT\_LOCAL + marginY;

    initSVG\_d(file,width,height);

    svg\_text\_d(file,loc,EAN13\_MAX\_HEIGHT\_LOCAL,EAN13\_TEXT\_SIZE\_D\_LOCAL,0,code[0]);

    loc+=EAN13\_FIRST\_SPACE\_LOCAL;

    // Add the first guard bar

    finalCode[0] = '1';

    finalCode[1] = '0';

    finalCode[2] = '1';

    svg\_line\_d(file,loc,marginY,loc,EAN13\_MAX\_HEIGHT\_LOCAL, EAN13\_WIDTH\_D\_LOCAL, 0);

    loc+=2\*EAN13\_WIDTH\_D\_LOCAL;

    svg\_line\_d(file,loc,marginY,loc,EAN13\_MAX\_HEIGHT\_LOCAL, EAN13\_WIDTH\_D\_LOCAL, 0);

    loc+=EAN13\_WIDTH\_D\_LOCAL;

    // Calculate the parity bit for the left part based on the first digit

    uint8\_t oddity = \_ean.oddity[code[0] - '0'];

    int index = 3; // Start index for the encoded left part

    // Encode the left part (6 digits)

    for(int i = 1; i < 7; i++) {

        uint8\_t digit = code[i] - '0';

        uint8\_t \*encodingSet = (oddity & (1 << (6 - i))) ? \_ean.A : \_ean.B;

        ///Print number

        svg\_text\_d(file,loc,EAN13\_MAX\_HEIGHT\_LOCAL,EAN13\_TEXT\_SIZE\_D\_LOCAL,0,code[i]);

        printf("I : %c ,:",code[i]);

        for (int j = 0; j < 7; j++) {

            finalCode[index++] = (encodingSet[digit] >> (6 - j)) & 1 ? '1' : '0';

            printf("%c", finalCode[index-1]);

            if(finalCode[index-1] == '1')

                svg\_line\_d(file,loc,marginY,loc,EAN13\_HEIGHT\_D\_LOCAL, EAN13\_WIDTH\_D\_LOCAL, 0);

            loc+=EAN13\_WIDTH\_D\_LOCAL;

        }

    }

    // Add the middle guard bar

    finalCode[index++] = '0';

    finalCode[index++] = '1';

    finalCode[index++] = '0';

    finalCode[index++] = '1';

    finalCode[index++] = '0';

    loc+=EAN13\_WIDTH\_D\_LOCAL;

    svg\_line\_d(file,loc,marginY,loc,EAN13\_MAX\_HEIGHT\_LOCAL, EAN13\_WIDTH\_D\_LOCAL, 0);

    loc+=2\*EAN13\_WIDTH\_D\_LOCAL;

    svg\_line\_d(file,loc,marginY,loc,EAN13\_MAX\_HEIGHT\_LOCAL, EAN13\_WIDTH\_D\_LOCAL, 0);

    loc+=2\*EAN13\_WIDTH\_D\_LOCAL;

    for(int i = 7; i < 13; i++) {

        uint8\_t digit = code[i] - '0';

        svg\_text\_d(file,loc,EAN13\_MAX\_HEIGHT\_LOCAL,EAN13\_TEXT\_SIZE\_D\_LOCAL,0,code[i]);

        for (int j = 0; j < 7; j++) {

            finalCode[index++] = (\_ean.A[digit] >> (6 - j)) & 1 ? '0' : '1';

            if(finalCode[index-1] == '1')

                svg\_line\_d(file,loc,marginY,loc,EAN13\_HEIGHT\_D\_LOCAL, EAN13\_WIDTH\_D\_LOCAL, 0);

            loc+=EAN13\_WIDTH\_D\_LOCAL;

        }

    }

    // Add the final guard bar

    finalCode[index++] = '1';

    finalCode[index++] = '0';

    finalCode[index++] = '1';

    svg\_line\_d(file,loc,marginY,loc,EAN13\_MAX\_HEIGHT\_LOCAL, EAN13\_WIDTH\_D\_LOCAL, 0);

    loc+=2\*EAN13\_WIDTH\_D\_LOCAL;

    svg\_line\_d(file,loc,marginY,loc,EAN13\_MAX\_HEIGHT\_LOCAL, EAN13\_WIDTH\_D\_LOCAL, 0);

    loc+=EAN13\_WIDTH\_D\_LOCAL;

    loc += 0.5;

    svg\_text\_d(file,loc,EAN13\_MAX\_HEIGHT\_LOCAL, EAN13\_TEXT\_SIZE\_D\_LOCAL,0,'>');

    endSVG(file);

    // Null-terminate the final code

    finalCode[index] = '\0';

    printf("Encoded EAN-13: %s\n", finalCode);

}

void drawCode(char\* filename, char\* encoded){

    FILE \*f = fopen(filename, "w");

    //Add some space around the code

    int marginX = 150;

    int marginY = 150;

    int width = strlen(encoded) \* EAN13\_WIDTH + 2 \* marginX;

    int height = EAN13\_HEIGHT + 2 \* marginY;

    initSVG(f, width, height);

    int i = 0 ;

    while(encoded[i]!='\0'){

        if(encoded[i] == '1')

        {

            // Add some space around the code

            svg\_line(f, marginX + i\*EAN13\_WIDTH, marginY, marginX + i\*EAN13\_WIDTH, marginY + EAN13\_HEIGHT, EAN13\_WIDTH, 0);

        }

        i++;

    }

    endSVG(f);

}

int main(){

    printf("Podaj kod: ");

    char code[] = "0634392677689";

    scanf("%s",code);

    while ((getchar()) != '\n');

    printf("Zmien skale [Y/n]: ");

    char x;

    scanf("%c",&x);

    while ((getchar()) != '\n');

    if(x=='Y' || x=='y'){

        printf("Nowa skala kodu: ");

        scanf("%d",&EAN13\_SCALE);

        while ((getchar()) != '\n');

    }

    char finalCode[100];

    encodeEan(code, finalCode);

    printf("Encoded EAN-13: %s\n", finalCode);

    char filename[] = "output.svg\0";

    drawCode(filename, finalCode);

}

# 3.Wnioski

Generacja kodów przebiega pomyślnie. Na zajęciach udało się podłączyć skaner kodów kreskowych który odczytywał poprawne przedmioty.

# 4. Źródła

[1] GS1 Strategy str 9 [https://www.gs1.org/docs/gs1-strategy-booklet.pdf](https://www.gs1.org/docs/gs1-strategy-booklet.pdf%20str9)

[2] <https://www.gs1.org/about/management-board>

[3] <https://mojegs1.pl/rejestracja/stworz-wniosek/gcp>

[4] <https://pl.wikipedia.org/wiki/GS1#cite_note-18>

[5] <https://mfiles.pl/pl/index.php/Kod_kreskowy>

[6] <https://pl.wikipedia.org/wiki/EAN>

[7] <https://romek.info/ut/barcode/ean13pl.html>

[8] Fragment tabeli <https://romek.info/ut/barcode/systemkodowy.html>

[9] <http://www.gernoth.net/rdf/ean13/ean13.html>

[10] <https://internationalbarcodes.com/ean-13-specifications>