

## Temă

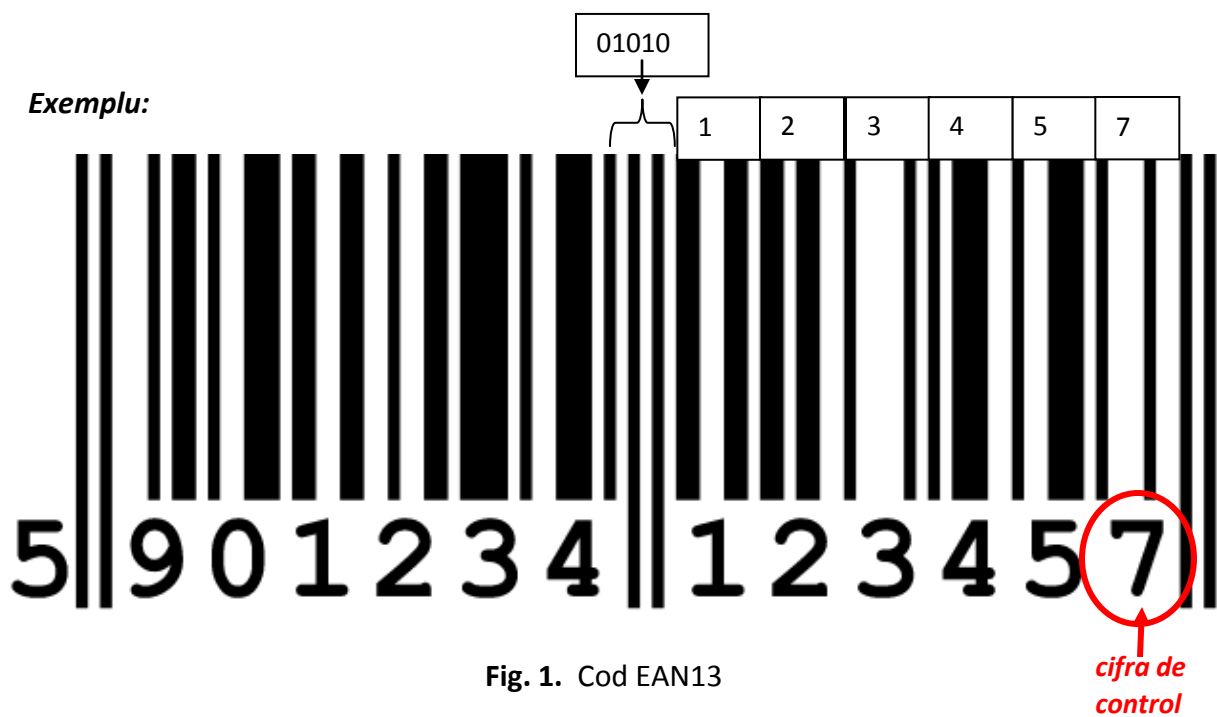
Se propune realizarea unei aplicații care să genereze coduri de bare conform standardului EAN13 (*European Article Number 13*).

### Bibliografie:

1. [http://en.wikipedia.org/wiki/International\\_Article\\_Number\\_\(EAN\)](http://en.wikipedia.org/wiki/International_Article_Number_(EAN))
2. <http://www.w3schools.com/svg/default.asp>
3. [http://en.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](http://en.wikipedia.org/wiki/Scalable_Vector_Graphics)

### Soluția va cuprinde:

1. Citirea unui cod format din maximum 9 cifre și a unei denumiri de produs (șir de caractere). Dacă s-a citit un cod mai scurt de 9 caractere se va completa codul citit până la nouă caractere prin adăugarea în fața secvenței citite a unor cifre '7'.
2. Calculul cifrei de control pentru un cod complet, de 12 cifre. Codul complet se obține adăugând în fața celor nouă obținute la punctul 1 a încă trei cifre care vor fi sistematic 594 (codul oficial al României).
3. Pentru codul completat la dreapta cu cifra de control (format deci din 13 cifre) se va scrie o funcție care generează codificarea binară echivalentă (95 cifre binare, 0 sau 1).  
Codificarea constă în :
  - a. codul de început (101, trasat cu bare mai lungi) +
  - b. 42 cifre binare ( $6 \cdot 7$ ) +
  - c. separatorul din mijloc (01010, trasat cu bare mai lungi),
  - d. 42 cifre binare ( $6 \cdot 7$ ) și
  - e. codul de sfârșit (101, trasat cu bare mai lungi).



Conform standardului [1] codul produsului și cifra de control (partea a doua a codului, 123457) se codifică folosind codurile din ultima coloană a tabelului 1. În exemplul dat, cele 42 cifre binare care trebuie adăugate în șirul de poziții binare vor fi:

1100110 1101100 1000010 1011100 1001110 1000100

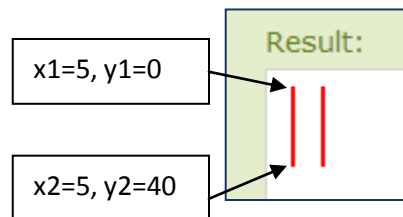
1 2 3 4 5 7

Tabelul 1

Digit	L-code	G-code	R-code
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

4. După generarea șirului de 95 de cifre binare se va scrie pe disc un fișier denumit *nume.html* (*nume* este numele produsului) conținând imaginea generată pe baza celor 95 de cifre binare, în cod *svg* (**S**calable **V**ector **G**raphics). Un exemplu de fișier care conține cod *svg* care trasează două linii este dat în continuare:

```
<!DOCTYPE html>
<html>
<body>
<svg height="50" width="200">
<line x1="5" y1="0" x2="5" y2="40" style="stroke:rgb(255,0,0);stroke-
width:2" />
<line x1="20" y1="0" x2="20" y2="40" style="stroke:rgb(255,0,0);stroke-
width:2" />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```



**Observații:**

1. La descrierea liniei, *rgb(255,0,0)* (*rgb* = **R**ed **G**reen **B**lue) stabilește codul culorii (roșu în exemplul dat). Pentru negru folosiți *rgb(0,0,0)*. Fiecare dintre cele trei componente ale culorii (roșu, verde și albastru) se exprimă printr-o valoare întreagă cuprinsă între 0 și 255.
2. Parametrul *stroke-width:2* indică lățimea linie trasate, în pixeli (puncte ecran). De acest parametru se va ține cont la stabilirea coordonatei *x* a barelor verticale. Barele corespund cifrelor '1' iar spațiile codifică valorile '0'.
3. La trasarea pe ecran, sensul pozitiv al axei *Ox* este spre dreapta iar sensul pozitiv al axei *Oy* este spre partea de jos a ecranului (*Oy este deci de sus în jos*).

***Succes!***