

**Przedmiot:**  
Układy Cyfrowe i Mikroprocesorowe

**Zadanie:**  
6. Symulator wyświetlacza  
alfanumerycznego.

Prowadzący: dr inż. Stanisław Lipski

Autor: Michał Bokinić, A1X1S1

# 1 Założenia:

- Wyświetlacz będzie się składał z 25 pikseli, jego wymiary to 5px x 5px.
- Piksele będą tworzyły znaki.
- Dane będą wpisywane z klawiatury.
- Piksele aktywne oznaczone zostaną znakiem 'X' a nieaktywne pustym miejscem.
- Program będzie działał w pętli a wyjście będzie powodowane wpisaniem niepoprawnego znaku.
- Poprawne znaki to litery, cyfry i znaki `": ; < = > ? @ "`.
- Każdy znak musi być opisany pikselami, tworzymy więc dla każdego znaku tekstowy obrazek 25x25px.
- Każda etykieta tekstowa będzie posiadała 26 znaków (25px i dolar jako znak końca linii), więc tyle też będzie wynosiła odległość między ich początkami w pamięci. Po zdefiniowaniu początku pierwszego znaku wyświetlacza w pamięci można dodać do niego wielokrotności liczby 26 otrzymując znaki kolejne.
- Należy także wprowadzić porównanie w celu sprawdzenia poprawności wpisywanych znaków.

Procedury wykorzystane w programie:

- funkcja 9 przerwania 21h - wyświetla tekst o adresie początkowym zdefiniowanym w dx
- funkcja 7 przerwania 21h - wczytuje znak do al bez echa
- funkcja 1 przerwania 21h - wczytuje znak do al z echem
- funkcja 0 przerwania 21h - kończy program

Etykiety wykorzystane w programie:

1. petla - służy do zapętlenia wprowadzania danych z klawiatury i wyświetlania znaków na wyświetlaczu alfanumerycznym, pętla ta kończona jest wprowadzeniem znaku innego niż dozwolone.
  - występuje w niej instrukcja `cmp` - porównania, która sprawdza, czy wpisany znak jest poprawny
  - występuje instrukcja `sub` - odejmowania, potrzebna do zamiany znaku ASCII na liczbę
  - występują instrukcje `mul` - mnożenia i `add` - dodawania, potrzebne, do określenia o ile znaków należy się przenieść w pamięci, aby wyświetlić na wyświetlaczu właściwy znak
  - występują także instrukcje `jmp`, `jg`, `jl` - skoku, odpowiednio: bezwarunkowy, oraz dwa zależne od wyniku porównania `cmp` ("większe" i "mniejsze")
2. stop - przenosi do instrukcji kończącej program, wywoływana jest w przypadku wpisania niepoprawnego znaku.

W programie występują także instrukcje:

- mov - kopiuje wartość drugiego parametru do parametru pierwszego
- int - wywołuje przerwanie
- push - wprowadza wartość na stos
- pop - odczytuje wartość ze stosu

*uzupełnienie opisu wykorzystanych etykiet, procedur oraz instrukcji*

## 2 Kod programu:

```
org 100h

push ax
push bx
push cx
push dx

mov ah, 9
mov dx, zaczynam
int 21h

znak = 26h ; tyle wynosi przesuniecie miedzy dwoma znakami wyswietlacza
mov bl, 30h ; kod ascii znaku '0'
mov bh, 7bh ; kod ascii znaku 'z'

petla:
    mov ah, 7h ; wczytuje znak bez echa
    int 21h ; znak w al

    cmp al, '0'
    jl stop ; sprawdzam czy znak wprowadzony jest mniejszy niz '0'
    cmp al, 'z'
    jg stop ; sprawdzam czy znak wprowadzony jest mniejszy niz 'z'

    sub al, '0' ; zamieniam znak ascii na liczbe porzadkowa znaku wyswietlacza
    mov dl, znak ; wrzucam przesuniecie do dl
    mul dl ; mnoze przesuniecie razy liczbe porzadkowa, aby wiedziec o ile znakow wyswietlacza mam sie
przeniesc do przodu

    mov dx, qq ; wrzucam do dx znak wyswietlacza - zero
    add dx, ax ; przenosze sie od zera o ilosc znakow wyswietlacza, aby wyswietlic ten wlasciwy

    mov ah, 9h ; wyswietlam znak wyswietlacza i zaczynam od nowa
    int 21h
jmp petla

stop:
mov ah, 9
mov dx, koncze
int 21h

mov ah, 1 ; czeka na klawisz
int 21h

pop ax
pop bx
pop cx
pop dx

mov ah, 0 ; ladne wyjscie z programu
int 21h

qq db 13, 10, ' xxx '
db 13, 10, 'x  x '
db 13, 10, 'x  x '
db 13, 10, 'x  x '
db 13, 10, 'xxx '
db 13, 10, '$'

qw db 13, 10, '  x '
db 13, 10, '  xx '
db 13, 10, ' x x '
db 13, 10, '  x '
db 13, 10, '  x '
db 13, 10, '$'

qe db 13, 10, ' xxx '
db 13, 10, 'x  x '
db 13, 10, '  xx '
db 13, 10, '  xx '
db 13, 10, 'xxxxx'
db 13, 10, '$'

qr db 13, 10, ' xxx '
db 13, 10, 'x  x '
db 13, 10, '  xx '
db 13, 10, 'x  x '
db 13, 10, 'xxx '
db 13, 10, '$'

qt db 13, 10, '  x '
db 13, 10, '  xx '
db 13, 10, ' x x '
db 13, 10, 'xxxxx'
db 13, 10, '  x '
db 13, 10, '$'

qy db 13, 10, 'xxxxx'
db 13, 10, 'x  '
db 13, 10, 'xxxx '
db 13, 10, '  x '
db 13, 10, 'xxxx '
db 13, 10, '$'
```

qu db 13,10,' xxx '  
db 13,10,' x '  
db 13,10,'xxxx '  
db 13,10,' x x '  
db 13,10,' xxx '  
db 13,10,' \$ '

qi db 13,10,'xxxxx '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' \$ '

qo db 13,10,' xxx '  
db 13,10,' x x '  
db 13,10,' xxx '  
db 13,10,' x x '  
db 13,10,' xxx '  
db 13,10,' \$ '

qp db 13,10,' xxx '  
db 13,10,' x x '  
db 13,10,' xxxx '  
db 13,10,' x '  
db 13,10,' xxx '  
db 13,10,' \$ '

qa db 13,10,' xx '  
db 13,10,' xx '  
db 13,10,' xx '  
db 13,10,' xx '  
db 13,10,' xx '  
db 13,10,' \$ '

qs db 13,10,' xx '  
db 13,10,' xx '  
db 13,10,' xx '  
db 13,10,' xx '  
db 13,10,' x '  
db 13,10,' \$ '

qd db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' \$ '

qf db 13,10,' '  
db 13,10,'xxxxx '  
db 13,10,' '  
db 13,10,'xxxxx '  
db 13,10,' '  
db 13,10,' \$ '

qg db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' \$ '

qh db 13,10,' xxx '  
db 13,10,' x x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' \$ '

qj db 13,10,' xxx '  
db 13,10,' x x '  
db 13,10,' xx x '  
db 13,10,' x xx '  
db 13,10,' xx '  
db 13,10,' \$ '

qk db 13,10,' xxx '  
db 13,10,' x x '  
db 13,10,'xxxxx '  
db 13,10,' x x '  
db 13,10,' x x '  
db 13,10,' \$ '

ql db 13,10,'xxxxx '  
db 13,10,' x x '  
db 13,10,'xxxxx '  
db 13,10,' x x '  
db 13,10,'xxxxx '  
db 13,10,' \$ '

qz db 13,10,' xxx '  
db 13,10,' x x '  
db 13,10,' x '  
db 13,10,' x x '  
db 13,10,' xxx '  
db 13,10,' \$ '

qx db 13,10,'xxxxx '  
db 13,10,' x x '  
db 13,10,' x x '  
db 13,10,' x x '  
db 13,10,'xxxxx '  
db 13,10,' \$ '

qc db 13,10,'xxxxx'  
db 13,10,'x'  
db 13,10,'xxxxx'  
db 13,10,'x'  
db 13,10,'xxxxx'  
db 13,10,'\$'

qv db 13,10,'xxxxx'  
db 13,10,'x'  
db 13,10,'xxxxx'  
db 13,10,'x'  
db 13,10,'x'  
db 13,10,'\$'

qb db 13,10,' xxx '  
db 13,10,'x '  
db 13,10,'x xxx '  
db 13,10,'x x '  
db 13,10,' xxx '  
db 13,10,'\$'

qn db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,'xxxxx '  
db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,'\$'

qm db 13,10,' xxx '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' xxx '  
db 13,10,'\$'

wq db 13,10,' xxxx '  
db 13,10,' x '  
db 13,10,' x '  
db 13,10,' x x '  
db 13,10,' xx '  
db 13,10,'\$'

ww db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,'xx '  
db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,'\$'

we db 13,10,'x '  
db 13,10,'x '  
db 13,10,'x '  
db 13,10,'x '  
db 13,10,'xxxxx '  
db 13,10,'\$'

wr db 13,10,'x x '  
db 13,10,'xx xx '  
db 13,10,'x x x '  
db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,'\$'

wt db 13,10,'x x '  
db 13,10,'xx x '  
db 13,10,'x x x '  
db 13,10,'x xx '  
db 13,10,'x x '  
db 13,10,'\$'

wy db 13,10,' xxx '  
db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,' xxx '  
db 13,10,'\$'

wu db 13,10,'xxxx '  
db 13,10,'x x '  
db 13,10,'xxxx '  
db 13,10,'x '  
db 13,10,'x '  
db 13,10,'\$'

wi db 13,10,' xxx '  
db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,'x xx '  
db 13,10,' xxx '  
db 13,10,'\$'

wo db 13,10,'xxxx '  
db 13,10,'x x '  
db 13,10,'xxxx '  
db 13,10,'x x '  
db 13,10,'x x '  
db 13,10,'\$'

```

wp db 13,10,' xxxx'
db 13,10,' x'
db 13,10,' xxx'
db 13,10,' x'
db 13,10,'xxxx'
db 13,10,'$'

wa db 13,10,'xxxxx'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,'$'

ws db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'xxx'
db 13,10,'$'

wd db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,' x'
db 13,10,'$'

wf db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x x'
db 13,10,'xx xx'
db 13,10,'x x'
db 13,10,'$'

wg db 13,10,'x x'
db 13,10,' x x'
db 13,10,' x'
db 13,10,' x x'
db 13,10,'x x'
db 13,10,'$'

wh db 13,10,'x x'
db 13,10,' x x'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,'$'

wj db 13,10,'xxxxx'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,'xxxxx'
db 13,10,'$'

wk db 13,10,' xx'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,' xx'
db 13,10,'$'

wl db 13,10,'x'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,'$'

wz db 13,10,' xx'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,' xx'
db 13,10,'$'

wx db 13,10,' x'
db 13,10,' x x'
db 13,10,'x x'
db 13,10,' x'
db 13,10,'$'

wc db 13,10,' '
db 13,10,' '
db 13,10,' '
db 13,10,' '
db 13,10,'xxxxx'
db 13,10,'$'

wv db 13,10,' xx'
db 13,10,' xx'
db 13,10,' x'
db 13,10,' x'
db 13,10,'$'

wn db 13,10,' xxx'
db 13,10,' x x'
db 13,10,'xxxxx'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'$'

```

```

wm db 13,10,'xxxx'
db 13,10,'x  x'
db 13,10,'xxxx'
db 13,10,'x  x'
db 13,10,'xxxx'
db 13,10,'$'

```

```

ew db 13,10,' xxx'
db 13,10,'x  x'
db 13,10,'x  '
db 13,10,'x  x'
db 13,10,' xxx'
db 13,10,'$'

```

```

ee db 13,10,'xxxx'
db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'xxxx'
db 13,10,'$'

```

```

er db 13,10,'xxxxx'
db 13,10,'x  '
db 13,10,'xxxxx'
db 13,10,'x  '
db 13,10,'xxxxx'
db 13,10,'$'

```

```

et db 13,10,'xxxxx'
db 13,10,'x  '
db 13,10,'xxxxx'
db 13,10,'x  '
db 13,10,'x  '
db 13,10,'$'

```

```

ey db 13,10,' xxx'
db 13,10,'x  x'
db 13,10,'x xxx'
db 13,10,'x  x'
db 13,10,' xxx'
db 13,10,'$'

```

```

eu db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'xxxxx'
db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'$'

```

```

ei db 13,10,' xxx'
db 13,10,'x  '
db 13,10,'x  '
db 13,10,'x  '
db 13,10,'xxx'
db 13,10,'$'

```

```

eo db 13,10,'xxxx'
db 13,10,'x  '
db 13,10,'x  '
db 13,10,'x  x'
db 13,10,'xx'
db 13,10,'$'

```

```

ep db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'xx'
db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'$'

```

```

ea db 13,10,'x  '
db 13,10,'x  '
db 13,10,'x  '
db 13,10,'x  '
db 13,10,'xxxxx'
db 13,10,'$'

```

```

ed db 13,10,'x  x'
db 13,10,'xx xx'
db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'$'

```

```

ef db 13,10,'x  x'
db 13,10,'xx x'
db 13,10,'x  x'
db 13,10,'x  xx'
db 13,10,'x  x'
db 13,10,'$'

```

```

eg db 13,10,' xxx'
db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'x  x'
db 13,10,'xxx'
db 13,10,'$'

```



```

eh db 13,10,'xxxx'
db 13,10,'x x'
db 13,10,'xxxx'
db 13,10,'x'
db 13,10,'x'
db 13,10,'$'

ej db 13,10,' xxx'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x xx'
db 13,10,'xxxx'
db 13,10,'$'

ek db 13,10,'xxxx'
db 13,10,'x x'
db 13,10,'xxxx'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'$'

el db 13,10,' xxxx'
db 13,10,'x'
db 13,10,' xxx'
db 13,10,'x'
db 13,10,'xxxx'
db 13,10,'$'

ez db 13,10,'xxxxx'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,'$'

ex db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'xxx'
db 13,10,'$'

ec db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x'
db 13,10,'$'

ev db 13,10,'x x'
db 13,10,'x x'
db 13,10,'x x x'
db 13,10,'xx xx'
db 13,10,'x x'
db 13,10,'$'

eb db 13,10,'x x'
db 13,10,' x x'
db 13,10,' x'
db 13,10,' x x'
db 13,10,'x x'
db 13,10,'$'

en db 13,10,'x x'
db 13,10,' x x'
db 13,10,' x'
db 13,10,' x'
db 13,10,'x x'
db 13,10,'$'

gq db 13,10,'xxxxx'
db 13,10,' x'
db 13,10,' x'
db 13,10,' x'
db 13,10,'xxxxx'
db 13,10,'$'

zaczniam db 13, 10, 9, 'witam w symulatorze wyswietlacza alfanumerycznego.'
db 13, 10, 9, 'wyswietlacz ma wymiary 5x5 pikseli.'
db 13, 10, 9, 'wprowadzony z klawiatury znak zostanie wyswietlony na wyswietlaczu.'
db 13, 10, 9, 'prosze wprowadzic cyfre, litere lub jeden z ponizszych znakow:',13,10,9,9,' : ; < = > ? @'
db 13, 10, 13, 10, 9, 'wprowadzenie innego znaku zakonczy program.$'
koncze db 13, 10, 13, 10, 9, 'wprowadzono znak spoza zakresu. Program zostanie zakonczony.'
db 13, 10,13,10, 9, 'Program napisal: Michal Bokiniec, AlXlSl.$'

```

### 3 Wnioski

Podczas pisania tego programu zauważyłem fakt, że wszystkie etykiety tekstowe wpisane do programu zostają dopisywane do pamięci jedna za drugą. Można zatem przeskoczyć niektóre etykiety aby dostać się do innych. Pozwoliło mi to wykorzystać proste operacje mnożenia i dodawania zamiast wpisywania instrukcji warunkowych dla każdego znaku oddzielnie. Miało więc to bardzo duży wpływ na długość głównej pętli programu.

Niestety nic nie dało się zrobić z częścią w której definiowałem tekst wyświetlany. Musiałem zaprojektować każdą literę oddzielnie, aby uzyskać obraz na symulowanym wyświetlaczu alfanumerycznym.

Aby to ominąć mógłbym wykorzystać wstawki C/C++ (np. poprzez komendę "*extrn*"), ale wtedy nie byłby to już czysty assembler.

Podsumowując: program działa, zadanie określone w tym ćwiczeniu zostało spełnione.