Příklad č. 1

Mikroprocesor ovládá systém průmyslové automatizace prostřednictvím příkazů (paketů), majících tento formát:

adresa	kód příkazu	data	kontrolní slovo

Každá část příkazu je 8 b slovo. Ta mají tento význam:

`adresa`: dle zařízení (0 – 31)

`kodPrikazu`: Určuje požadovanou operaci. Např. 3 = proveď manipulaci s výstupem č. X

`data`: obsah se liší dle kódu příkazu. Pro kód 3:
bity b0 – b2 → kód manipulace

Kód	Akce
0	Vypni výstup
1	Sepni výstup
2	Periodicky (f = 58 min^{-1}) spínej výstup
3	Periodicky (f = 104 min^{-1}) spínej výstup
4	Na dobu 200 ms sepni výstup

bity b3 – b7 → číslo výstupu X (0 – 31)

`kontrolniSlovo`: `adresa XOR hlavicka XOR data`

V JSA napište program, který z proměnných `adresa`, `kodPrikazu`, `kodManipulace` a `cisloVystupu` sestaví daný paket (pro kód příkazu 3) – určete tedy hodnoty slov `data` a `kontrolniSlovo`. Výsledný paket umístěte do paměti jako pole `paket`.

```
          .include "m32def.inc"

// konstanty
.equ      N         = 4      // Délka paketu v B

// aliasy registrů
.def      rData     = r0
.def      rTemp     = r1
.def      rXor      = r2

// proměnné
          .DSEG
// Hodnoty budou přiřazeny ručně v AVR studiu
adresa:   .byte     1
kPrik:    .byte     1
kManip:   .byte     1
cisloV:   .byte     1

paket:    .byte     N

// program
          .CSEG
          .ORG      0x0000

          // Vytvoření slova "data"
          lds       rData, kManip
          lds       rTemp, cisloV
          lsl       rTemp          // cisloV << 3
          lsl       rTemp
```

```
            lsl         rTemp
            or          rData, rTemp   // Bitové sečtení -> vytvoření data

            // Vytvoření "paketu"
            ldi         YL, LOW(paket)
            ldi         YH, HIGH(paket)
            lds         rTemp, adresa
            st          Y, rTemp       // paket[0] = adresa
            lds         rTemp, kPrik
            std         Y+1, rTemp     // paket[1] = kPrik
            std         Y+2, rData     // paket[2] = rData
            // Výpočet kontrolního slova
            ld          rXor, Y+       // rXor = paket[0] = adresa
            ld          rTemp, Y+      // rTemp = paket[1] = kPrik
            eor         rXor, rTemp    // adresa ^ kPrik
            ld          rTemp, Y+      // rTemp = paket[2] = data
            eor         rXor, rTemp    // rXor = adresa ^ kPrik ^ data
            st          Y, rXor        // paket[3] = rXor = kontrolniSlovo
end:        rjmp        end            // Nekonečná smyčka
```