Lab 4 - procedury

```
; Program odczytujący z klawiatury napis i wypisujący go.

org 100h

        mov     ah, 9                   ; funkcja wyświetlania na ekran
        mov     dx, jak_masz
        int     21h

        mov     ah, 0ah         ; funkcja pobierania danych z klawiatury
        mov     dx, imie
        int     21h                     ; pobierz dane

        mov     ah, 9
        mov     dx, powitanie
        int     21h                     ; wyświetl napis "Cześć"

        mov     ah, 9
        mov     dx, imie+2      ; adres wpisanych danych
        int     21h

        mov     ax, 4Ch
        int     21h

        jak_masz        db      "Jak masz na imie? $"
        imie            db      20              ; maksymalna liczba znaków do pobrania
                        db      0               ; tu dostaniemy, ile znaków pobrano
                        times 22 db "$"         ; miejsce na dane

        powitanie       db      10, 13, 10, 13, "Czesc $"
```

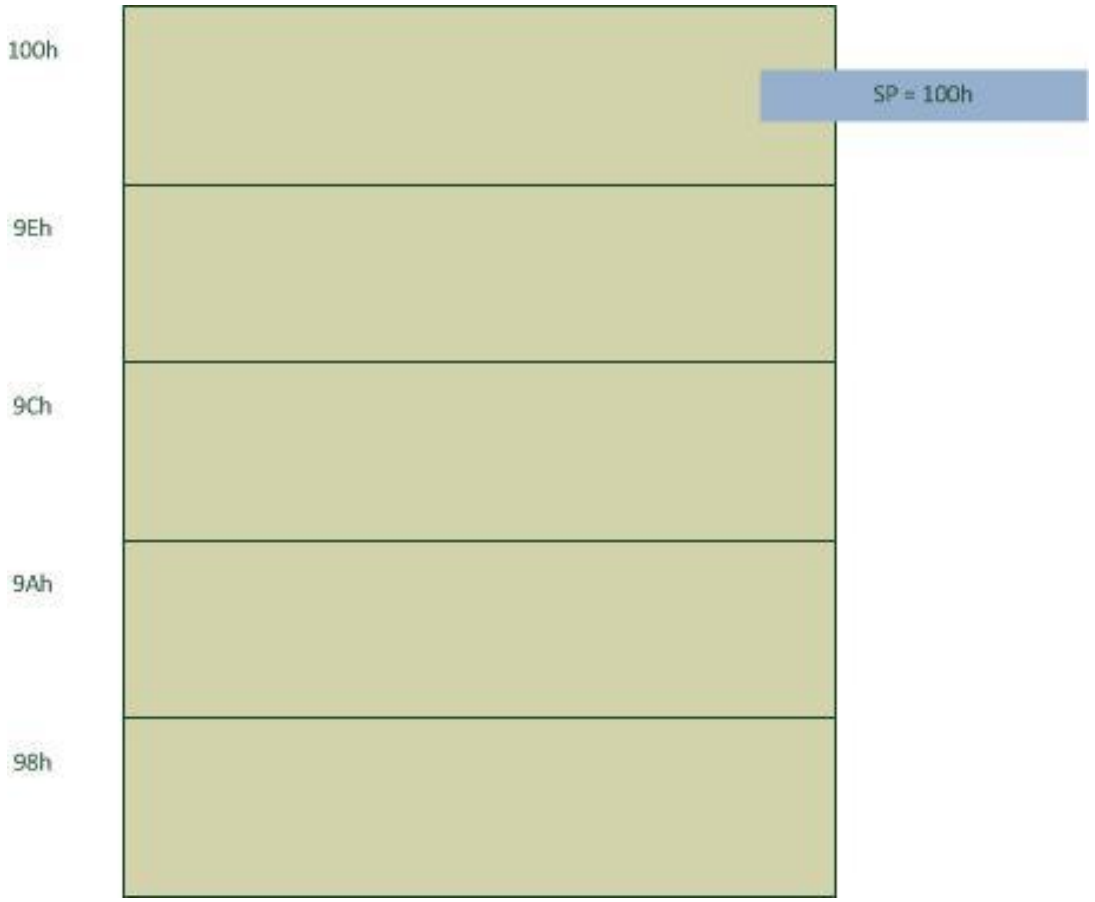
- Procedura: = część kodu (zwykle jeden kompletny algorytm), który istnieje w całym programie tylko w jednym miejscu, ale może być w nim wykonywany dowolną ilość razy i w dowolnym jego miejscu
- Kluczowe instrukcje:
  - call (skok do procedury, zapisanie adresu powrotu na stosie)
  - ret (powrót z procedury, pobranie adresu ze stosu)

```
;przykład programu wykonującego procedurę. Należy go
    uruchomić i sprawdzić, co robi.

org 100h

start:
    call procedure
    mov ah, 4Ch
    int   21h


procedure:
    mov ax, 9
    mov bx, 11
    add ax, bx
    ret
```

- Zadanie:

  Napisać program, który w pętli będzie prosił o wprowadzenie napisu z klawiatury i wyświetlał go na ekranie. Należy wykorzystać procedury

Lab 5 - stos

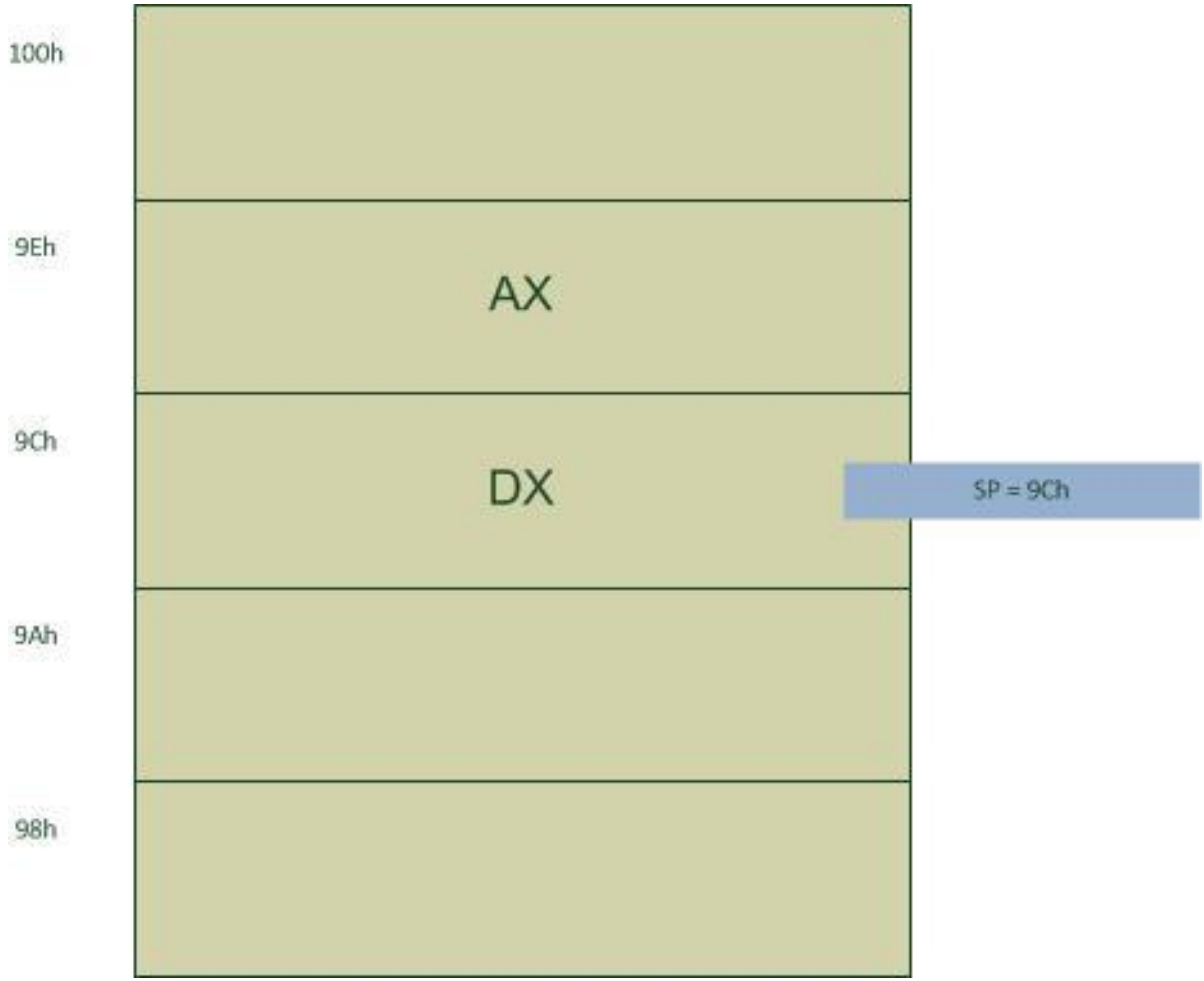


Rejestry:

- SS – przechowuje adres stosu
- SP – adres wierzchołka stosu
- BP – adres danych w segmencie stosu
- IP – adres aktualnie wykonywanej instrukcji

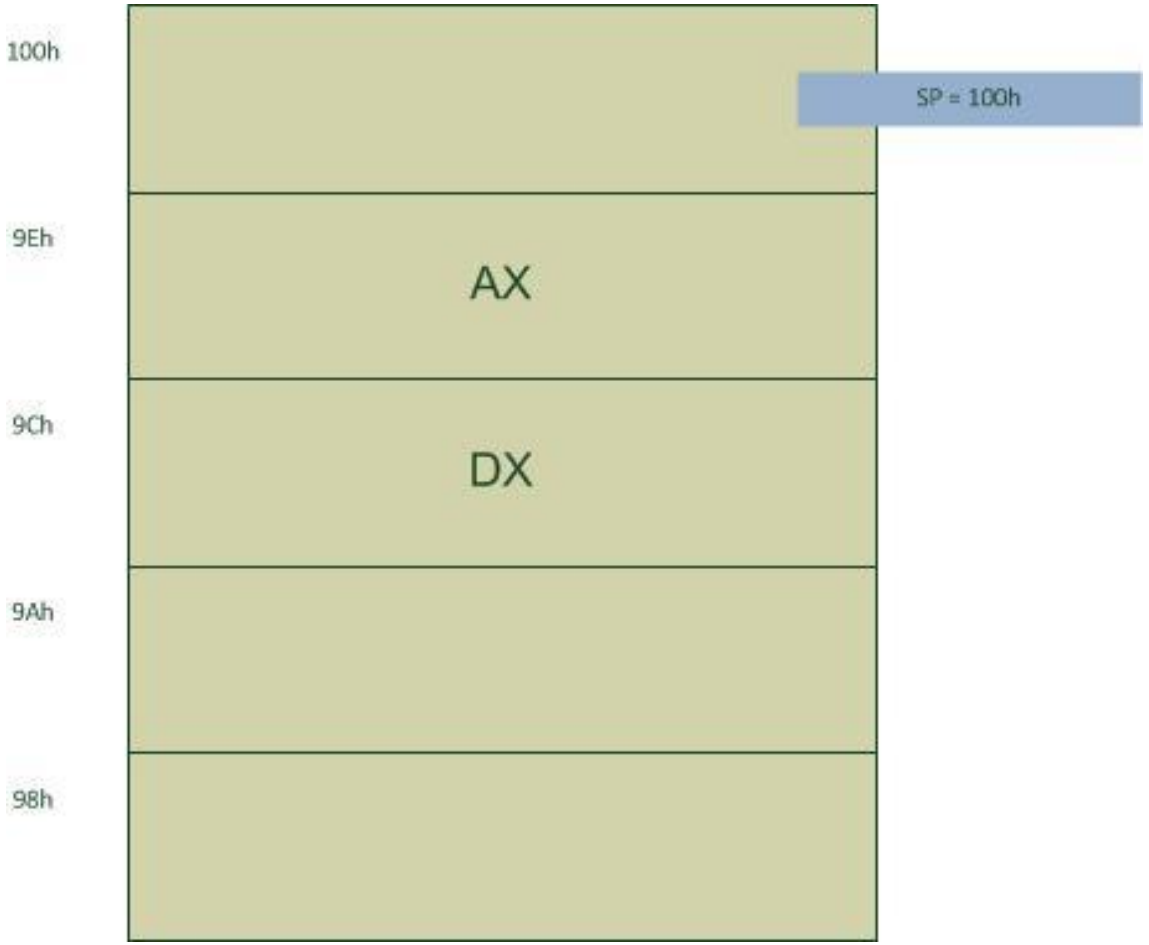
push AX

push DX



pop DX

pop AX



```
; pod koniec procedury SP musi być taki sam, jak na
    początku – wersja poprawiona
procedure2:
    push ax
    push bx
    add ax, bx

    add sp, 8

ret ; na wierzchu stosu jest bx, a nie adres powrotny
    z procedury
```

```
; rezerwacja miejsca na argumenty
push    bp              ; zachowanie starego BP
mov     bp, sp          ; BP = SP

sub     sp, xxx         ; rezerwacja miejsca na zmienne lokalne
push    rej1            ; tu SP się zmienia, ale BP już nie
push    rej2
   ...
pop     rej2            ; tu SP znów się zmienia, a BP - nie
pop     rej1

mov     sp, bp          ; zwalnianie zmiennych lokalnych

pop     bp

ret
```

- Zadanie:

  Napisać program, który wykorzystując stos i procedurę pozwoli dodać do siebie dwie cyfry i wyświetli wynik na ekranie

```
org 100h

mov cx, 8

start:

mov ah, 07h
int 21h
sub ax, 30h

mov bx, 2

push ax
push bx
call sumuj

mov ah,02h
mov dl, al
add dl, 30h
int 21h

loop start

mov ah, 4ch
int 21h
```

```
sumuj:

push bp

mov bp, sp ; nie można użyć pop - na stosie adres
      ;powrotu z wywołania call

mov ax, [bp + 6] ; ładowanie pierwszej liczby
mov bx, [bp + 4] ; ładowanie drugiej liczby

add ax,bx ; sumowanie

pop bp
ret
```