Nell'esercizio di oggi ci viene chiesto di esaminare il codice assembly di seguito ed identificare i costrutti C equivalenti. Inoltre provare a ipotizzare la funzionalità del codice.

```
.text:00401000                push    ebp
.text:00401001                mov     ebp, esp
.text:00401003                push    ecx
.text:00401004                push    0               ; dwReserved
.text:00401006                push    0               ; lpdwFlags
.text:00401008                call    ds:InternetGetConnectedState
.text:0040100E                mov     [ebp+var_4], eax
.text:00401011                cmp     [ebp+var_4], 0
.text:00401015                jz      short loc_40102B
.text:00401017                push    offset aSuccessInterne ; "Success: Internet Connection\n"
.text:0040101C                call    sub_40105F
.text:00401021                add     esp, 4
.text:00401024                mov     eax, 1
.text:00401029                jmp     short loc_40103A
.text:0040102B ; ---------------------------------------------------------------------------
.text:0040102B
```

Nel codice si possono identificare tre costrutti. Un ciclo if, la creazione dello stack nella memoria e il passaggio dei parametri alla funzione.
Nella figura sotto viene evidenziato il ciclo if:

```
.text:00401000                push    ebp
.text:00401001                mov     ebp, esp
.text:00401003                push    ecx
.text:00401004                push    0               ; dwReserved
.text:00401006                push    0               ; lpdwFlags
.text:00401008                call    ds:InternetGetConnectedState
.text:0040100E                mov     [ebp+var_4], eax
.text:00401011                cmp     [ebp+var_4], 0
.text:00401015                jz      short loc_40102B
.text:00401017                push    offset aSuccessInterne ; "Success: Internet Connection\n"
.text:0040101C                call    sub_40105F
.text:00401021                add     esp, 4
.text:00401024                mov     eax, 1
.text:00401029                jmp     short loc_40103A
.text:0040102B ; ---------------------------------------------------------------------------
.text:0040102B
```

L’istruzione “jz” esegue un salto alla locazione specificata quando lo zero flag è uguale a 1, ovvero quando il risultato della sottrazione è uguale a 0. Quindi l’istruzione “cmp” serve a capire se la variabile 4 è uguale a 0.
La funzione di questo codice sembra essere quella di capire se c’è una connessione disponibile con la quale raggiungere internet. Se la condizione del ciclo non è soddisfatta verrà stampato a schermo “Success: Internet Connection” altrimenti verrà effettuato il salto a una locazione 40102B riportata più in basso.

```
.text:00401000                push    ebp
.text:00401001                mov     ebp, esp
.text:00401003                push    ecx
.text:00401004                push    0               ; dwReserved
.text:00401006                push    0               ; lpdwFlags
.text:00401008                call    ds:InternetGetConnectedState
.text:0040100E                mov     [ebp+var_4], eax
.text:00401011                cmp     [ebp+var_4], 0
.text:00401015                jz      short loc_40102B
.text:00401017                push    offset aSuccessInterne ; "Success: Internet Connection\n"
.text:0040101C                call    sub_40105F
.text:00401021                add     esp, 4
.text:00401024                mov     eax, 1
.text:00401029                jmp     short loc_40103A
.text:0040102B ; ---------------------------------------------------------------------------
.text:0040102B
```

Nell’immagine sopra viene evidenziato la parte di codice in cui viene creato lo stack della funzione, mentre in quella sotto il passaggio dei parametri alla funzione tramite il comando “push”.