

## [Sessione 3] Controllo di flusso

### [s3.1] Numero pari successivo

*nome del file sorgente: parisuccessivo.asm*

Si scriva il codice che dato un intero inserito dall'utente restituisca il numero pari successivo.

### [s3.2] Check condizioni

*nome del file sorgente: checkCondizioni.asm*

Si scriva il codice assembly che esegua le seguenti istruzioni:

```
a = <intero inserito dall'utente>
b = <intero inserito dall'utente>
c = <intero inserito dall'utente>

If ( (a>=b) && (c!=0) ){
    z=c(a+b);
    print z
}
else{
    print «errore»
}
```

### [s3.3] Somma quadrati

*nome del file sorgente: sommaQuadrati.asm*

Si scriva il codice che calcola la somma dei primi N-1 numeri elevati al quadrato. Nel caso in cui l'i-esimo numero da aggiungere sia multiplo del valore iniziale della somma, si termini il ciclo for.

```
V=<intero inserito dall'utente>;
N=<intero inserito dall'utente>;

Sum = V;
for (i=1; i<N; i++)
{
    If ((i*i)%V==0){
        print «break»;
        break;
    }
}
```

```
Sum+=i*i;  
}  
print Sum
```

### [s3.4] Manipolazione di un array

*nome del file sorgente: rvarray.asm*

Si scriva il codice assembly che:

- inizializzi il segmento dati con un array di 13 interi scelti a piacere;
- chieda all'utente di inserire tre interi  $a$ ,  $b$  e  $c$ ;
- se  $c=0$ , scambi l' $a$ -esimo elemento dell'array con il  $b$ -esimo;
- se  $c=1$ , sovrascriva il  $b$ -esimo elemento dell'array con il valore dell' $a$ -esimo elemento;
- se  $c=-1$ , sovrascriva l' $a$ -esimo elemento dell'array con il valore del  $b$ -esimo elemento;
- se  $c$  ha un altro valore stampi la stringa «comando non riconosciuto» e non effettui modifiche all'array.