

```
procedure sum( $a_1, a_2, \dots, a_n$  integers)
```

```
  sum :=  $a_1$ 
```

```
   $i := 2$ 
```

```
  while  $i \leq n$ 
```

```
    sum := sum +  $a_i$ 
```

```
     $i := i + 1$ 
```

1. Can a_n be empty? What is i at the end of **while**?
2. Pre-conditions?
3. Post-conditions?
4. Loop invariant?

```
procedure find( x integer, a1, a2, ..., an integers)
  i := n
  while ( i > 0 and x <> ai )
    i--
  return i
```

5. Determine the number of times each statement is executed for the above algorithm, in the best case execution.
6. Determine the number of times each statement is executed for the following algorithm, in the worst case.
7. Give an algorithm that takes as input a list of n integers and returns the index of the *last* even integer in the list or 0 if there are no even integers.

```
procedure sum( $a_1, a_2, \dots, a_n$  integers)
```

```
  sum :=  $a_1$ 
```

```
   $i := 2$ 
```

```
  while  $i \leq n$ 
```

```
    sum := sum +  $a_i$ 
```

```
     $i := i + 1$ 
```

1. Can a_n be empty? No. i at the end of **while**? $n+1$

2. Pre-conditions? $\exists a_1$

3. Post-conditions? $\forall j: 1 \leq j \leq n, \text{sum} := \sum a_j$

4. Loop invariant? $\forall j: 1 \leq j < i, \text{sum} := \sum a_j$

4. Determine the number of times each statement is executed for the following algorithm, in the worst case.

procedure find(x integer, a ₁ , a ₂ , a ₃ ,..., a _n integers)	Best	Worst
i:=n	1	1
while (i > 0 and x <> a _i)	1	n+1
i--	0	n
return i	1	1

5. Determine the number of times each statement is executed for the above algorithm, in the best-case execution.

6. Describe an algorithm that takes as input a list of n integers and returns the index of the *last* even integer in the list or 0 if there are no even integers.

```
procedure last(  $a_1, a_2, a_3, \dots, a_n$  )  
   $i := n$   
  while (  $i > 0$  and  $a_i \bmod 2 = 1$  )  
     $i--$   
  return  $i$ 
```