have some characteristics in common, but also each of them has different way of organizing the data elements/objects they contain).

Collection is a container of discrete values;
Usually of the same type (primitive data values and also some other data structures);
(But) collection objects can be of different types (pointers afford a flexibility and thus collection objects permit references to any data structure as well as to primitive values);
Collections have a set of methods that define operations performed on the elements/objects of that collection;
Such as adding/removing elements to/from collection, comparing elements of collection, searching, *etc*;
Which reduces programming effort (because implementations of data structures and algorithms are provided);
Which increases performance of the program (because efficient implementations are provided); **[3]**

(d) *Award **[6 max]**.*

*Award **[1]** for initialization and for outputting **correct** result (COUNTER).*
*Award **[1]** for using collection methods.*
*Award **[1]** for correct loop.*
*Award **[1]** for retrieving a number (`ELEMENT`) from the collection.*
*Award **[1]** for if statement within the loop.*
*Award **[1]** for correct condition in if statement.*
*Award **[1]** for increasing `COUNTER` if needed.*

Example answer:

```
COUNTER = 0
NUMBERS.resetNext()
loop while NUMBERS.hasNext()
  ELEMENT = NUMBERS.getNext()
  if ELEMENT >= -1 and ELEMENT <= 1 then // abs(ELEMENT) <= 1
    COUNTER = COUNTER + 1
  end if
end loop
output COUNTER
```

***Note:** be flexible over the method names. For example, NUMBERS.getData() is acceptable instead of `NUMBERS.getNext()`.* **[6]**