// Integration test plan

//TEST CASE 1: memory is a power of 2 and greater than minimum chunk size

```
-4
//*  TO TESTER:
          check for error message on console */
3
//*  TO TESTER:
          check for error message on console */
653
//*  TO TESTER:
          check for error message on console */
256
//*  TO TESTER:
          verify that 256 has been accepted */
```

//TEST CASE 2: correct file size

```
1
error
0
//*  TO TESTER:
          check for error message on console */
1
error
-6
//*  TO TESTER:
          check for error message on console */
1
error
278
//*  TO TESTER:
          check for error message on console */
```

//TEST CASE 3: memory maxed-out

```
1
file1
256
1
file2
4
```

```
//*  TO TESTER:
          check for error message on console */
5   //clear memory
1
file1
4
1
file2
128
3
//* TO TESTER:
          check memory output on screen.
          view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: 1 --- File Size: 4
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 8  --- empty ---
3:  Chunk Size: 16  --- empty ---
4:  Chunk Size: 32  --- empty ---
5:  Chunk Size: 64  --- empty ---
6:  Chunk Size: 128 Data Name: 2 --- File Size: 128
*/
1
file3
128
//*  TO TESTER:
          check for error message on console */
1
file3
65
//*  TO TESTER:
          check for error message on console */
1
file3
64
3
```

```
//* TO TESTER:
        check memory output on screen.
        view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: 1 --- File Size: 4
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 8  --- empty ---
3:  Chunk Size: 16  --- empty ---
4:  Chunk Size: 32  --- empty ---
5:  Chunk Size: 64 Data Name: 3 --- File Size: 64
6:  Chunk Size: 128 Data Name: 2 --- File Size: 128
*/
```

//TEST CASE 4: test clear function

```
5    //clear memory
3
//* TO TESTER:
        check memory output on screen.
        view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 256   --- empty ---
*/
```

//TEST CASE 5: split empty memory into smallest chunk

```
1
SmallFile
4
3
//* TO TESTER
        check memory output on screen.
        view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: SmallFile --- File Size: 4
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 8  --- empty ---
3:  Chunk Size: 16  --- empty ---
4:  Chunk Size: 32  --- empty ---
5:  Chunk Size: 64  --- empty ---
6:  Chunk Size: 128  --- empty ---
*/
```

//TEST CASE 6: merge memory form smallest chunk to largest

```
2
SmallFile
3
//* TO TESTER:
            check memory output on screen.
            view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 256   --- empty ---
*/
```

//TEST CASE 7: files placed in smallest possible chunk

```
1
file1
4
1
file2
16
3

//* TO TESTER:
            check memory output on screen.
            view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: 1 --- File Size: 4
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 8  --- empty ---
3:  Chunk Size: 16 Data Name: 2 --- File Size: 16
4:  Chunk Size: 32  --- empty ---
5:  Chunk Size: 64  --- empty ---
6:  Chunk Size: 128  --- empty ---
*/
1
file3
16
3
//* TO TESTER:
            check memory output on screen.
            view should look like this:
```

```
The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: 1 --- File Size: 4
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 8  --- empty ---
3:  Chunk Size: 16 Data Name: 2 --- File Size: 16
4:  Chunk Size: 16 Data Name: 3 --- File Size: 16
5:  Chunk Size: 16  --- empty ---
6:  Chunk Size: 64  --- empty ---
7:  Chunk Size: 128  --- empty ---
*/
5     //clear memory
1
file1
1
1
file2
5
1
file3
9
1
file4
17
1
file5
33
3
//* TO TESTER:
            check memory output on screen.
            view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: file1 --- File Size: 1
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 8 Data Name: file2 --- File Size: 5
3:  Chunk Size: 16 Data Name: file3 --- File Size: 9
4:  Chunk Size: 32 Data Name: file4 --- File Size: 17
5:  Chunk Size: 64 Data Name: file5 --- File Size: 33
6:  Chunk Size: 128  --- empty ---
*/
```

//TEST CASE 8: clear memory

```
4


//* TO TESTER:
        check memory output on screen.
        view should look like this:

Wasted Memory: 59
Available Memory: 132
*/
5    //clear memory
```

//TEST CASE 9: only buddies merge

```
//populates memory

1
file1
4
1
file2
4
1
file3
4
1
file4
4
1
file5
4
1
file6
4
1
file7
4
1
file8
4

//begin test

2
file2
2
file3
2
file5
2
```

```
file7
3
//* TO TESTER:
            check memory output on screen.
            view should look like this:

file7The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: file1 --- File Size: 1
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 4  --- empty ---
3:  Chunk Size: 4 Data Name: file4 --- File Size: 4
4:  Chunk Size: 4  --- empty ---
5:  Chunk Size: 4 Data Name: file6 --- File Size: 4
6:  Chunk Size: 4  --- empty ---
7:  Chunk Size: 4 Data Name: file8 --- File Size: 4
8:  Chunk Size: 32  --- empty ---
9:  Chunk Size: 64  --- empty ---
10:  Chunk Size: 128  --- empty ---
*/
2
file6
3
//* TO TESTER:
            check memory output on screen.
            view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: file1 --- File Size: 4
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 4  --- empty ---
3:  Chunk Size: 4 Data Name: file4 --- File Size: 4
4:  Chunk Size: 8  --- empty ---
5:  Chunk Size: 4  --- empty ---
6:  Chunk Size: 4 Data Name: file8 --- File Size: 4
7:  Chunk Size: 32  --- empty ---
8:  Chunk Size: 64  --- empty ---
9:  Chunk Size: 128  --- empty ---
*/
2
file6
3
```

```
//* TO TESTER:
        check memory output on screen.
        view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 4 Data Name: file1 --- File Size: 4
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 8  --- empty ---
3:  Chunk Size: 8  --- empty ---
4:  Chunk Size: 4  --- empty ---
5:  Chunk Size: 4 Data Name: file8 --- File Size: 4
6:  Chunk Size: 32  --- empty ---
7:  Chunk Size: 64  --- empty ---
8:  Chunk Size: 128  --- empty ---
*/
2
file1
3
//* TO TESTER:
        check memory output on screen.
        view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 16  --- empty ---
1:  Chunk Size: 8  --- empty ---
2:  Chunk Size: 4  --- empty ---
3:  Chunk Size: 4 Data Name: file8 --- File Size: 4
4:  Chunk Size: 32  --- empty ---
5:  Chunk Size: 64  --- empty ---
6:  Chunk Size: 128  --- empty ---
*/
2
file8
3
//* TO TESTER:
        check memory output on screen.
        view should look like this:

The current size of the memory system is 256
0:  Chunk Size: 256   --- empty ---
*/
```

//TEST CASE10: resize memory

```
6
3
45
```

```
//*  TO TESTER:
            check for error message on console */
-5
//*  TO TESTER:
            check for error message on console */
64
//* TO TESTER:
            check memory output on screen.
            view should look like this:

The current size of the memory system is 64
0:  Chunk Size: 64  --- empty ---
*/
1
file1
2
3
//* TO TESTER:
            check memory output on screen.
            view should look like this:

The current size of the memory system is 64
0:  Chunk Size: 4 Data Name: 1 --- File Size: 2
1:  Chunk Size: 4  --- empty ---
2:  Chunk Size: 8  --- empty ---
3:  Chunk Size: 16  --- empty ---
4:  Chunk Size: 32  --- empty ---
*/
5     //clear memory
3
//* TO TESTER:
            check memory output on screen.
            view should look like this:

The current size of the memory system is 64
0:  Chunk Size: 64  --- empty ---
*/


exit    //end testing
```