The following contains the runtimes of Workloads A-D plus the extra workloads we have created (E, F), repeated 10 times. The runtimes are given in microseconds.

All the workloads were implemented as intended and were repeated 100 times.

Our metadata consisted of one unsigned short (2 bytes = 16 bits) divided into 2 parts: 4 bits were allocated for denoting that the block was free (isFree = 1 or 0), and the rest of the 12 bits were to denote the size of the block. We went with 12 bits because our max size was 4096, and 2^12 = 4096.

Our metadata struct was stored adjacent to the specific block of allocated memory about which it refers to. This way, we are able to return a pointer to the allocated memory just by iterating over the size of our metadata.

For our mymalloc function, we wrote two separate functions, one that finds available memory of the required size or larger, and one that splits the memory block if it is greater than the one required. It then sets the extra memory to free and returns a pointer to the memory just allocated.

Our myfree function iterates over the memory blocks using the size of each block stored in our metadata for each block. Once it finds the target address, it checks the previous and next memory block and merges them together if they are free. Otherwise, it just sets the target memory block to free.

Run 1

```
totalA: 539.000000

totalB: 3183.000000

totalC: 593.000000

totalD: 635.000000

totalE: 1666.000000

totalF: 1193.000000

meanA: 5.390000

meanB: 31.830000

meanC: 5.930000

meanD: 6.350000

meanE: 16.660000

meanF: 11.930000
```

Run 2

```
totalA: 619.000000

totalB: 3617.000000

totalC: 692.000000

totalD: 690.000000

totalE: 1917.000000

totalF: 1352.000000

meanA: 6.190000

meanB: 36.169998

meanC: 6.920000

meanD: 6.900000

meanE: 19.170000

meanF: 13.520000
```

Run 3

```
totalA: 618.000000

totalB: 3595.000000

totalC: 691.000000

totalD: 687.000000

totalE: 1929.000000

totalF: 1336.000000

meanA: 6.180000

meanB: 35.950001

meanC: 6.910000

meanD: 6.870000

meanE: 19.290001

meanF: 13.360000
```

Run 4

```
totalA: 448.000000

totalB: 2653.000000

totalC: 518.000000

totalD: 511.000000

totalE: 1393.000000

totalF: 1047.000000

meanA: 4.480000

meanB: 26.530001

meanC: 5.180000

meanD: 5.110000

meanE: 13.930000

meanF: 10.470000
```

Run 5

```
totalA: 590.000000

totalB: 3479.000000

totalC: 689.000000

totalD: 693.000000

totalE: 1831.000000

totalF: 1379.000000

meanA: 5.900000

meanB: 34.790001

meanC: 6.890000

meanD: 6.930000

meanE: 18.309999

meanF: 13.790000
```

Run 6

```
totalA: 537.000000

totalB: 3088.000000

totalC: 593.000000

totalD: 594.000000

totalE: 1647.000000

totalF: 1166.000000

meanA: 5.370000

meanB: 30.879999

meanC: 5.930000

meanD: 5.940000

meanE: 16.469999

meanF: 11.660000
```

Run 7

```
totalA: 621.000000

totalB: 3606.000000

totalC: 674.000000

totalD: 703.000000

totalE: 1924.000000

totalF: 1348.000000

meanA: 6.210000

meanB: 36.060001

meanC: 6.740000

meanD: 7.030000

meanE: 19.240000

meanF: 13.480000
```

Run 8

```
totalA: 611.000000
totalB: 3593.000000
totalC: 696.000000
totalD: 696.000000
totalE: 1899.000000
totalF: 1334.000000
meanA: 6.110000
meanB: 35.930000
meanC: 6.960000
meanD: 6.960000
meanE: 18.990000
meanF: 13.340000
```

Run 9

```
totalA: 635.000000
totalB: 3634.000000
totalC: 701.000000
totalD: 708.000000
totalE: 1910.000000
totalF: 1424.000000
meanA: 6.350000
meanB: 36.340000
meanC: 7.010000
meanD: 7.080000
meanE: 19.100000
meanF: 14.240000
```

```
Run 10
totalA: 616.000000
totalB: 3620.000000
totalC: 682.000000
totalD: 699.000000
totalE: 1910.000000
totalF: 1339.000000
meanA: 6.160000
meanB: 36.200001
meanC: 6.820000
meanD: 6.990000
meanE: 19.100000
meanF: 13.390000
```

RUN	TOTAL A	TOTAL B	TOTAL C	TOTAL D	TOTAL E	TOTAL F
1	539	3183	593	635	1666	1193
2	619	3617	792	690	1917	1352
3	618	3595	691	687	1929	1336
4	448	2653	518	511	1393	1047
5	590	3479	689	693	1831	1379
6	537	3088	593	594	1647	1166
7	621	3606	674	703	1924	1348
8	611	3593	696	696	1899	1334
9	635	3634	701	708	1910	1424
10	616	3620	682	699	1910	1339
MEAN	583.4	3406.8	662.9	661.6	1802.6	1291.8
MED	613.5	3594	685.5	691.5	1904.5	1337.5
PER TRIAL						
MEAN	5.834	34.068	6.629	6.616	18.026	12.918

The results vary quite a bit after consecutive trials, but we can see that Workload B has the largest runtime (34.068 usec per trial), trailed by E (18.026 usec) and F (12.918 usec).

We discovered that even though the workloads were random for C and D, their runtimes were similar and smaller than what we expected. As for F, we were surprised to see that using strcpy had a longer runtime.