

```
main.c x + ... >_ Console x Shell x +
main.c > ...
22
23   for (int i = my_start; i < my_end; i++) {
24       A[i / n][i % n] = rand() % 100; // Números inteiros aleatórios entre 0 e 99 (ajuste
conforme necessário)
25   }
26
27   return NULL;
28 }
29
30 int main() {
31     // Defina os valores de m, n, thread_count e outros parâmetros, se necessário
32     m = 3; // 3 linhas na matriz
33     n = 3; // 3 colunas na matriz
34     thread_count = 2; // Número de threads
35
36     // Aloque espaço para armazenar identificadores das threads
37     pthread_t* thread_handles = malloc(thread_count * sizeof(pthread_t));
38
39     for (long thread = 0; thread < thread_count; thread++) {
40         pthread_create(&thread_handles[thread], NULL, GenerateRandomNumbers, (void*) thread);
41     }
42
43     for (long thread = 0; thread < thread_count; thread++) {
44         pthread_join(thread_handles[thread], NULL);
45     }
46
47     free(thread_handles);
48
49     // Imprima a matriz A com formatação
50     printf("Matriz A:\n");
51     for (int i = 0; i < m; i++) {
52         for (int j = 0; j < n; j++) {
53             printf("%02d ", A[i][j]); // Formatação para dois dígitos com zero à esquerda
54         }
55         printf("\n");
56     }
57
58     return 0;
59 }
Ln 59, Col 2 • Spaces: 2 History
```

```
> make -s
> ./main
Matriz A:
71 54 41
63 71 54
41 63 86
```

```
main.c x + ... >_ Console x Shell x +
main.c > ...
22
23   for (int i = my_start; i < my_end; i++) {
24       A[i / n][i % n] = rand() % 100; // Números inteiros aleatórios entre 0 e 99 (ajuste
conforme necessário)
25   }
26
27   return NULL;
28 }
29
30 int main() {
31     // Defina os valores de m, n, thread_count e outros parâmetros, se necessário
32     m = 3; // 3 linhas na matriz
33     n = 3; // 3 colunas na matriz
34     thread_count = 2; // Número de threads
35
36     // Aloque espaço para armazenar identificadores das threads
37     pthread_t* thread_handles = malloc(thread_count * sizeof(pthread_t));
38
39     for (long thread = 0; thread < thread_count; thread++) {
40         pthread_create(&thread_handles[thread], NULL, GenerateRandomNumbers, (void*) thread);
41     }
42
43     for (long thread = 0; thread < thread_count; thread++) {
44         pthread_join(thread_handles[thread], NULL);
45     }
46
47     free(thread_handles);
48
49     // Imprima a matriz A com formatação
50     printf("Matriz A:\n");
51     for (int i = 0; i < m; i++) {
52         for (int j = 0; j < n; j++) {
53             printf("%02d ", A[i][j]); // Formatação para dois dígitos com zero à esquerda
54         }
55         printf("\n");
56     }
57
58     return 0;
59 }
Ln 59, Col 2 • Spaces: 2 History
```

```
> make -s
> ./main
Matriz A:
97 87 22
46 97 87
22 46 72
```