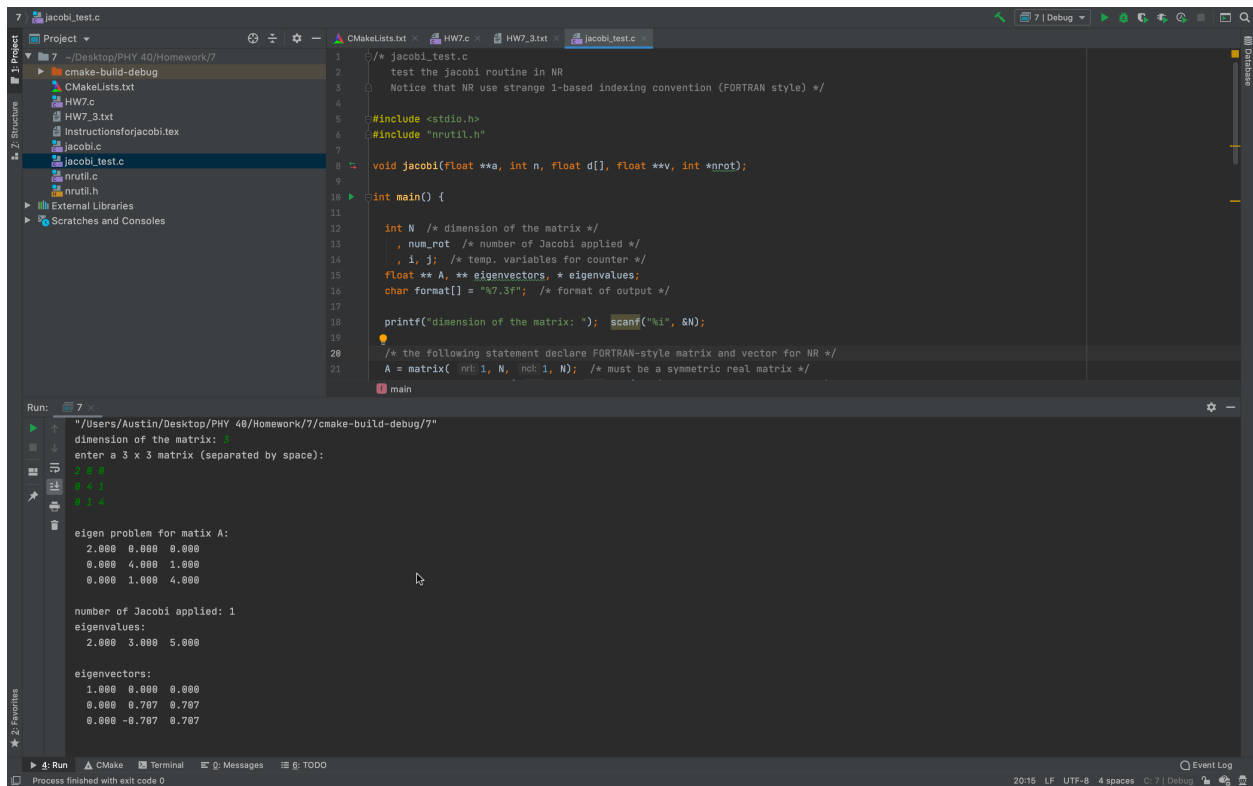


[HW7-1]



```
1  /* jacobi_test.c
2  test the jacobi routine in NR
3  Notice that NR use strange 1-based indexing convention (FORTRAN style) */
4
5  #include <stdio.h>
6  #include "nrutil.h"
7
8  void jacobi(float **a, int n, float d[], float **v, int *nrot);
9
10 int main() {
11
12     int N /* dimension of the matrix */
13     , num_rot /* number of Jacobi applied */
14     , i, j; /* temp. variables for counter */
15     float **A, **eigenvectors, *eigenvalues;
16     char format[] = "%7.3f"; /* format of output */
17
18     printf("dimension of the matrix: "); scanf("%i", &N);
19
20     /* the following statement declare FORTRAN-style matrix and vector for NR */
21     A = matrix( nril:1, N, nril:1, N); /* must be a symmetric real matrix */
22
23     main
24 }
```

Run: 7

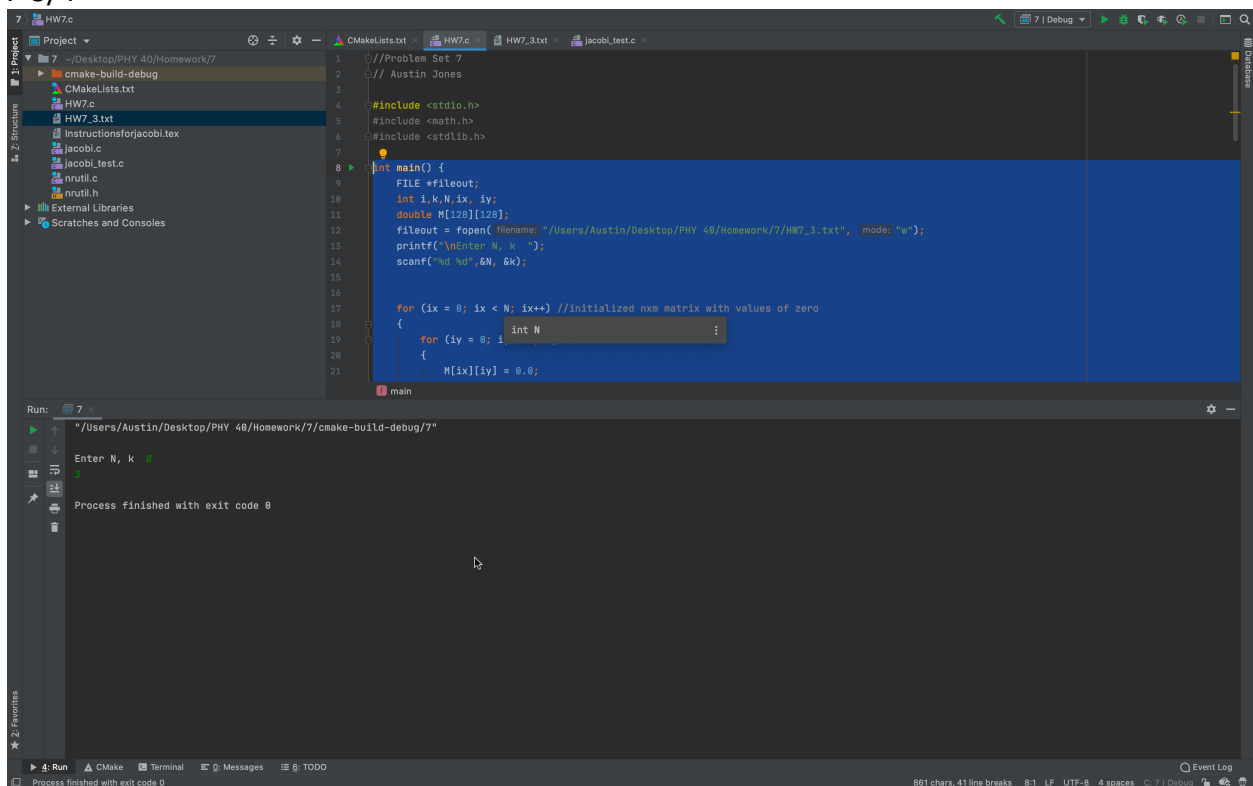
```
"/Users/Austin/Desktop/PHY 40/Homework/7/cmake-build-debug/7"
dimension of the matrix:
enter a 3 x 3 matrix (separated by space):
2.000 0.000 0.000
0.000 4.000 1.000
0.000 1.000 4.000

eigen problem for matrix A:
2.000 0.000 0.000
0.000 4.000 1.000
0.000 1.000 4.000

number of Jacobi applied: 1
eigenvalues:
2.000 3.000 5.000

eigenvectors:
1.000 0.000 0.000
0.000 0.707 0.707
0.000 -0.707 0.707
```

7-3/4



```
1  //Problem Set 7
2  // Austin Jones
3
4  #include <stdio.h>
5  #include <math.h>
6  #include <stdlib.h>
7
8  int main() {
9
10     FILE *fileout;
11     int i,k,N,ix,iy;
12     double M[128][128];
13     fileout = fopen( filename: "/Users/Austin/Desktop/PHY 40/Homework/7/HW7_3.txt", mode: "w");
14     printf("\nEnter N, k ");
15     scanf("%d %d",&N, &k);
16
17     for (ix = 0; ix < N; ix++) //initialized nxm matrix with values of zero
18     {
19         for (iy = 0; iy < N; iy++)
20         {
21             M[ix][iy] = 0.0;
22         }
23     }
24 }
```

Run: 7

```
"/Users/Austin/Desktop/PHY 40/Homework/7/cmake-build-debug/7"
Enter N, k
Process finished with exit code 0
```