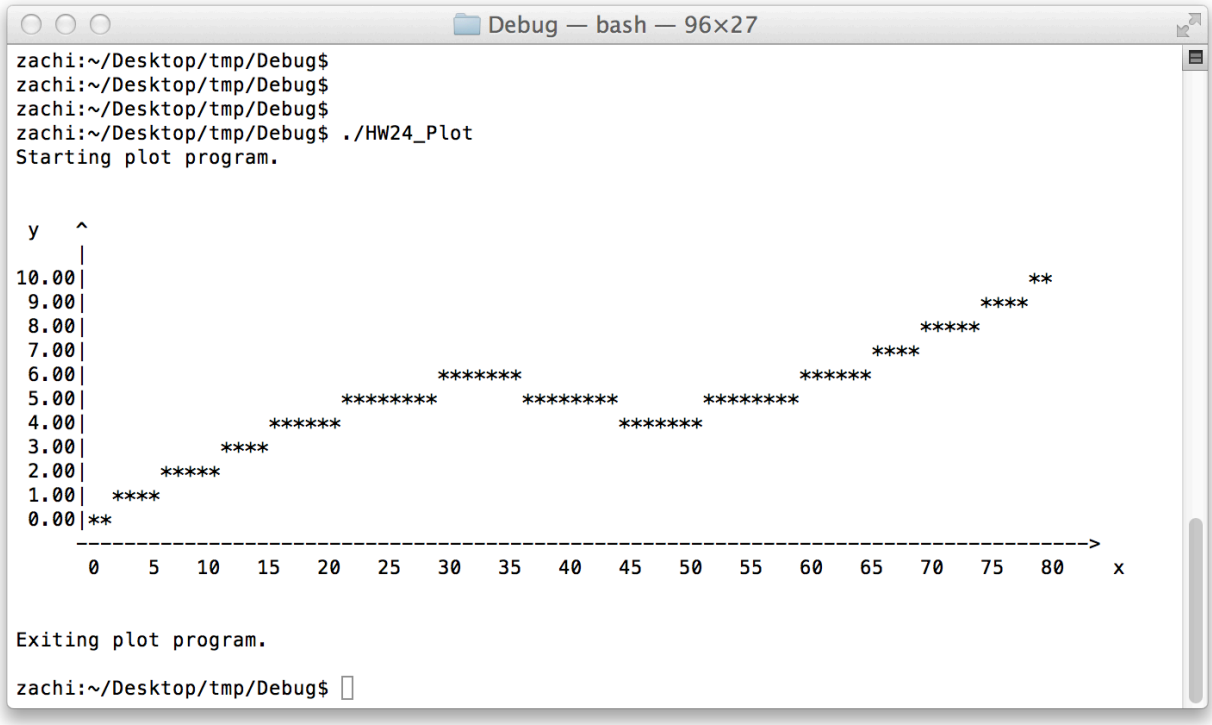
Assignment HW24
Due date (on or before): Announced in class.

1. Plot a graph

In this exercise you will plot a given function using ASCII art.
Below is a sample plot of a given function rendered by the program.



Please use the skeleton code given below, and fill-in the code in the noted place.

If you really want: Create a different function to plot.

Skeleton code:

```
//
//  main.c
//  HW24_Plot
//
//
```

```c
#include <stdio.h>
#include <math.h>


#define PTS 80

// Plot:
// void plot(double x[], double y[], int l)
// x[], y[] - arrays of values
// l - length of arrays
// For this plot program, assume:
// y-range is from 0 to 10.
// y-resolution: 1
// x-range is from 0 to (l-1).
// x-resolution: 1


void plot(double x[], double y[], int l)
{

}


int main()
{
    double yVal[PTS]=
{0,0,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,5,5,5,6,6,6
,6,6,6,6,5,

5,5,5,5,5,5,5,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,6,6,6,6,6,6,7,7,7,7,
8,8,8,8,8,9,9,9,9,10,10};
    // If you are wondering about these values, see below how
they were computed.
    double xVal[PTS];
    int ii;

    printf("Starting plot program.\n\n\n");

    for (ii=0; ii<PTS; ++ii) {
        xVal[ii] = ii;
        //yVal[ii] = round(10*fabs( sin((ii/40.0+0.25/2)*2*M_PI)
));
    }

    plot(xVal,yVal,PTS);
```

```
    printf("Exiting plot program.\n");
    getchar();

    return (0);
}

// Y values:
/*    Pseudocode (Matlab)

 x=linspace(-1,1,80);

 a=0.2;
 gamma=1.5;

 y= (abs(x-a).^gamma-a^gamma).*(x>0) - (abs(x+a).^gamma-
a^gamma).*(x<0);

 maxy = max(y);
 miny=min(y);

 y = round( (y-miny)/(maxy-miny)*10 );

 figure;
 plot(0:79,y,'*');

 printf('%d,',y);
 */
```

********** Start of code + Screen shot**********

********** End of code + Screen shot**********

2. 2D arrays, functions: Calculate cofactors

{ General knowledge (just FYI): The cofactor of a matrix is the determinant of the matrix resulting from eliminating the row and column of the specific element. }

In this exercise, you are given a 3x3 2D array of integers, and you are required to calculate the cofactor-matrix (up to a sign) , which is another 3x3 matrix, and it's elements are given as below:

```
                                  0   -1   -2
Given matrix    : A[3][3] =       1    0   -1
                                  4    3    2
```

Calculate Cofactor matrix C[3][3] in the following manner:

Procedure to calculate C[i][j]. For example, we'll take C[0][1].
1. Remove row i and column j from A ==> create new matrix A2

We remove row 0 and Column 1:

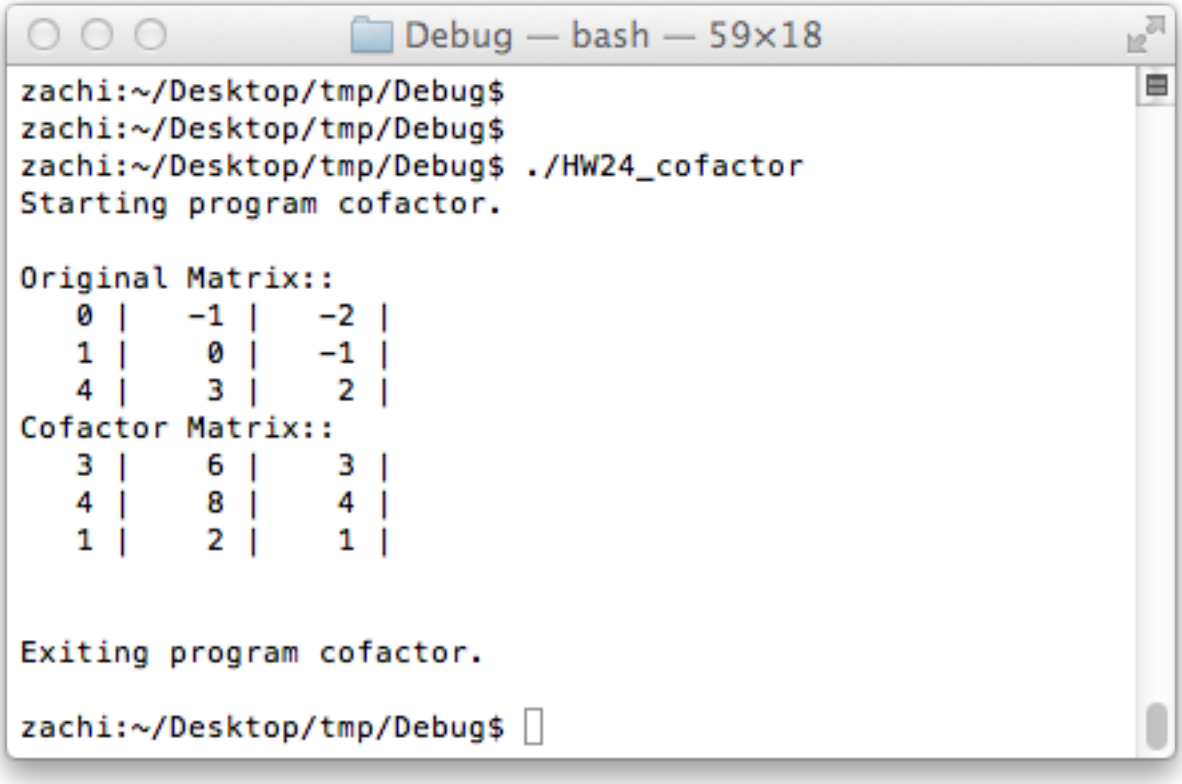
```
          A2  = 1  -1
                4   2
```

2. Calculate the diagonals-products, and take the difference.

Result = 1*2 – (-1)*4 = 2 + 4 = 6

3. Assign the result to C[i][j].
C[0][1] = 6

See screenshot below, and try to calculate some of the values on your own by hand, to make sure you understand the computation process.



Now, fill-in the code below to produce the above results.

```
//
//  main.c
//  HW24_cofactor
//
//

#include <stdio.h>


int cofactor()
{

     // fill-in code here, including input parameters.
     return(0);
}

void printMatrix(int M[3][3])
```

```
{

     // fill-in code here, to print an input matrix

}

int main()
{

    int A[3][3];
    int B[3][3];
    int rr,cc;

    printf("Starting program cofactor.\n\n");

    for (rr=0; rr<3; ++rr)
        for (cc=0; cc<3; ++cc)
            A[rr][cc]= rr*rr-cc;

    printf("Original Matrix::\n");
    printMatrix(A);


     // fill-in code here, to calculate B.
    // You probably want to create a loop on the leemnts of B,
    // and call for each the function cofactor, which you will
    // write above.

    printf("Cofactor Matrix::\n");
    printMatrix(B);

    printf("\n\nExiting program cofactor.\n");

    getchar();
    return(0);
}
```

********** Start of code + Screen shot**********

********** End of code + Screen shot**********