Please write neatly and show all of your work. Adequate space for your response has been provided following each question. If you need additional room, turn to the backside of the page or ask the instructor for paper. Good luck!!

<table>
<thead>
<tr>
<th>Question</th>
<th>Possible Score</th>
<th>Actual Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Extra credit</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>136</td>
<td></td>
</tr>
</tbody>
</table>
1) **(10 points)** Write the screen display for the following script and associated function in the space provided.

% This is part of a MATLAB script

\[
a = 5 \\
b = 4 \\
[a,b] = confuse(b,a)
\]

The following script is stored in a file named confuse.m

... function [b,a] = confuse(a,b)  
    a = 3*a  
    b = 2*b-a

<table>
<thead>
<tr>
<th>Display #</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

2) **(15 points)** Show the output generated by the following program:

```
#include<stdio.h>

main( )
{
    int x=2, y=2, k=1;

    do
    {
        y = y * x;
        printf("k = %d  y = %d \n", k, y);
        y = y - k;
        printf("k = %d  y = %d \n", k, y);
        k = k + 1;
    }
    while (k < 4);
}
```
3) (15 points) What will be displayed by the following program? Pay attention to formatting in the printf statement.

```c
#include<stdio.h>
main()
{ /* begin main */
    int a = -3 , b = 3, c = -2, d = 5, e, f;
    double alfa = 2.333, beta = 2.0, gamma = 5.3, delta, epsilon;

    delta = (a*b/c)*gamma;
    e = d%b;
    epsilon = (alfa*beta/c)*b;
    beta = (1/2)*beta+d;
    f = c*d+c-a/d+d;

    printf( "\ndelta = %5.3f 
e = %d 
epsilon = %.2f"
        "\nbeta = %.1f\nne = %d \nepsilon = %.2f"
    "\n", delta, e, epsilon, beta, f);
} /* end main */
```

delta =
e =
epsilon =
beta =
f =
YourName:

4. (10 points) The following program compiles with no error messages but produces a run time error. The student that wrote this program did not completely understand the relationship between declared variable types and input/output functions.

a) What caused the error?

b) After this problem is corrected the program displays incorrect results when executed. Why?

#include<stdio.h>
#define PI 3.14159
main()
{ /* begin main */
    double radiusi, radiuso, area;

    printf("Please enter value for inner radius> ");
    scanf("%lf", radiusi);

    printf("Please enter value for outer radius> ");
    scanf("%lf", radiuso);

    area = PI*(radiuso*radiuso – radiusi*radiusi);

    printf("An annulus with an inner radius %d and outer radius %d has an area of %d", radiusi, radiuso, area);
} /* end main */
5) (18 points) The following three blocks contain portions of code from a C program.

In the first Box to the right of the code, describe in one sentence the purpose of the section of code, i.e. identify the algorithm.

In the lower empty box to the right of each block of code, enter the primary Matlab command, or function, which performs the equivalent task to the C code shown (For example, if the code were performing a linear regression using the minimization of squares algorithm, you would put “polyfit” in the box for the Matlab command)

```c
#define SIZE 10
int a[ ] = {2, 51, 14, 8, 10, 12, 89, 68, 45, 37};
int i, pass, hold;
for (pass = 1;   pass < 10;   pass++)
{
   for (i = 0;   i < SIZE - 1;   i++)
   {
      if (a[i] > a[i + 1])
      {
         hold = a[i];
         a[i] = a[i + 1];
         a[i + 1] = hold;
      }
   }
   for (i = 0;  i < 10;  i++)  printf("%d, ", a[i]);
```

Equivalent Matlab command:

Describe purpose of code:

```c
float xlow, xhi, step, area = 0;
int i, j, no_divs;

printf("Enter the x axis range: \n");
scanf("Low limit = %f \nUpper limit = %f \n", &xlow, &xhi);
scanf("Enter number of divisions to use: %d\n", &no_divs);

step = (xhi - xlow) / no_divs;
for (i = 0; i <= no_divs; i++)
{
   x[i] = xlow + i * step;
   y[i] = function_of_x(x[i]);
}
for (j = 0; j < no_divs; j++)
   area = area + (step * (y[j] + y[j + 1]) / 2);
printf(\Result is: %f \n", area);
```

Describe purpose of code:

Equivalent Matlab command:
6) (34 points) The following program is in need of some prototypes

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

main( )
{
    int a, b, c;
    double f, g, h;
    double many[20], lots[20];
    char ans, name[21];

    displayheader();

    do {
        a = getdata( name, &f, many );
        h = calculate( f, many );
        manipulate( a, f, &g, many, lots );
        c = errorcheck( f, g, lots );
        report( name, c, lots );
        ans = goagain();
        while ( toupper(ans) = 'Y' );
    } // end main
```

Describe purpose of code:

Equivalent Matlab command:

```matlab
desk = zeros(size, size);
desk = desk + 1;
for k = 1:size
    x(k) = data(k);
    y(k) = data(k + size/2);
end
fclose(fptr);
```
YourName: 

a) Write an appropriate prototype for each of the functions used in the following lines:

Line 11: 

Line 14: 

Line 15: 

Line 16: 

Line 17: 

Line 18: 

Line 19: 

b) You have been asked to convert the preceding C program to a MATLAB script with corresponding function .m files. Write the first line of the function .m file for the functions used in the following lines of the C program above.

Line 15: 

Line 18:
For the following C program, answer the questions on the following page:

```c
#include<stdio.h>
void CrossProd(double a[], double b[], double c[]);
void DisplayMsg(int flag, char message[]);

void main(void) {
    char msg[] = "\n\nFinished!! Have a Nice Summer. \n";
    double v1[] = {1.0, 0.0, 1.0};
    double v2[] = {1.0, 2.0, 0.0};
    double out[3];
    int i;

    DisplayMsg(1, " ");
    DisplayMsg(0, "Computing Cross Product \n");

    CrossProd(v1, v2, out);

    DisplayMsg(0, "\nResult: \n");
    for (i=0;i<3;i++)
        printf("\n  C[%i] = %4.1lf", i, out[i]);

    DisplayMsg(0, msg);
    return;
}

void CrossProd(double y[], double z[], double a[]) {
    a[0] = y[1]*z[2] - y[2]*z[1];
    a[1] = y[2]*z[0] - y[0]*z[2];
    a[2] = y[0]*z[1] - y[1]*z[0];
    return;
}

void DisplayMsg(int welcome_flag, char message[]) {
    if (welcome_flag ==1 )
        printf("\nWelcome to my Program \nI hope you enjoy it! \n\n");
    else
        printf("%s", message);

    return;
}
```
a) (24 pts) Write the screen display from running the programs (including any blank lines!).

<table>
<thead>
<tr>
<th>Display #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
</tbody>
</table>

b) (10 pts) Rewrite the function `CrossProd(double y[], double z[], double a[])` to run in MATLAB, using the same input and output variable names.