

Sample

Midterm Exam 1

COMPUTER PROGRAMMING FOR ENGINEERING AND SCIENCE
Held on 23th July, 2010 (CSCE 150E, SUMMER 2010)

Name :
Course No : CSCE150E

3a) give just 1 reason
3c) m, not p
7c) simplify first
9b) ==, not =

Instructions:

1. This is open book, open note, but not open neighbor.
2. If you have a question about the meaning of an exercise, ask! Getting things wrong because of misunderstandings can be aggravating for me as well as you.
3. If answers do not fit in the space provided, use the back of a sheet and very carefully indicate and label this so I don't miss it in grading.
4. You may take the entire period.

1. (10 points) Keeping in mind that a logical array is utilized quite differently from a numeric array, consider the follow interaction:

```
>> x = [3 5 1 7 6 2 4]
x =
     3     5     1     7     6     2     4
>> y = x > 0
y =
     1     1     1     1     1     1     1
>> z = double(y)
z =
     1     1     1     1     1     1     1
```

- (a) What is the result of entering the following two expressions?

3

```
>> x(y)    3 5 1 7 6 2 4
>> x(z)    3 3 3 3 3 3 3
```

- (b) How many bytes are used to store each of the three variables?

3

```
x : 7 * 8 = 56
y : 7 * 1 = 7
z : 7 * 8 = 56
```

- (c) The above interaction could be automatically preserved in a disk file. How?

2

```
save filename diary filename
```

- (d) The workspace created above could be saved to a disk file and restored into a later Matlab run. How?

2

```
save filename
load filename
```

2. (10 points) The following code is designed to do digit by digit multiplication.

(a) What does m represent?

2

2-D array of partial products

(b) Show the contents of m at the end of the first loop. Be careful! (You don't need to show all of it - just enough to make sure you have the idea.)

2

$$m = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & & \\ & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 16 & 40 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 14 & 35 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 6 & 15 \end{bmatrix}$$

(c) Show the contents of p at the end of the second loop.

2

$$p = [0 \ 0 \ 0 \ 0 \ 8 \ 7 \ 19 \ 54 \ 41 \ 15]$$

(d) What is missing after the second loop?

2

adjust p to be all single digits, (^{process} carries)

(e) What does t represent?

2

the number of characters printed

```
a = [0 0 0 0 0 0 1 0 2 5];
b = [0 0 0 0 0 0 0 8 7 3];
s = length(a);
m = zeros(s, s);
for ii = 1:s
    m(ii, :) = b(ii) * a;
end
p = zeroes(1, s);
for ii = 1:s
    p(1:ii) = p(1:ii) + m(ii, s-ii+1:s);
end
% something is missing here
f = 'The product is %s\n';
t = fprintf(f, p)
```

3. (10 points) Consider the workspace following execution of the previous exercise.

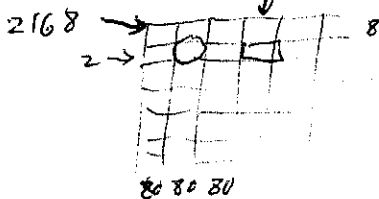
(a) Why is m preassigned to 0's? Give two reasons.

save time (instead of copy + expand cycle)

(b) Assuming the variables are stored in the workspace in sequence and in chronological order, and the beginning of storage is at memory address 2000, what are the beginning addresses of each variable?

		size	offset	addr
a	10 * 8	80	0	2000
b	10 * 8	80	80	2080
s	8	8	160	2160
m	10 * 10 * 8	800	168	2168
li	8	8	968	2968
p	10 * 8	80	976	2976
f	19 * 2	38	1056	3056
t	8	8	1094	3094

(c) Using the same assumptions, what are the starting addresses of $m(2,4)$ and $m(12)$?



$$\text{offset} = 3 * 80 + 8 = 248$$

$$2168 + 248 = 2416$$

$$\text{offset} = 80 + 8 = 88$$

$$2168 + 88 = 2256$$

(d) If the workspace is saved to a file, it MIGHT consume the same amount of space. Explain my use of the word 'MIGHT'.

-ascii designator would convert internal binary to presentation ASCII which might not consume the same bytes

4. (10 points) Consider the following code.

(a) What is the final value of count in the following code segment?

5

424

(b) Give a formula for the final value to show how it can be calculated.

5

```

count = 0;
for a = 1:4
    for b = 1:6
        count = count + 1;
    end
    for c = 1:2
        for d = 1:5
            for e = 1:3
                count = count+1;
            end
            for f = 1:7
                count = count+1;
            end
        end
    end
end
end
end
end

```

$$4 * (6 + 2 * 5 * (3 + 7))$$

$$\frac{\frac{10}{100}}{100}$$

424

5. (10 points) What is output by the following fragment of code?

10

```

for a = 3:-1:1
    for b = a:3
        for c = 3:-1:b
            fprintf('%1d %1d %1d\n', a, b, c)
        end
        fprintf('har! har! har!\n')
    end
    fprintf('giggle!\n')
end
end

```

3 3 3 har
giggle!

2 2 2 har

2 3 3 har
giggle!

1 1 2

1 1 1 har

1 2 3

1 2 2 har

1 3 3 har
giggle!

6. (10 points) Consider the following function definition. Write another (complete) function that uses it to compute the volume of a sphere. It should use a single assignment statement and contain no operators.

7 if
no
fn

```
function x = mOp(a, b, op)
switch op
case '+'
    x = a + b;
case '-'
    x = a - b;
case '*'
    x = a * b;
case '/'
    x = a / b;
case '^'
    x = a ^ b;
otherwise
    x = 0;
end
```

$$\left(\left(\left(\frac{4}{3} \right) * \pi \right) * (r_b^3) \right) - \left(\left(4 * \pi \right) * (r_m^3) \right)$$

$$4\pi \left(\frac{r_b^3}{3} - r_m^3 \right)$$

function ~~Vol~~ volume(r)

$$V = \text{mOp}(\text{mOp}(\text{mOp}(4, 3, '/'), \pi, '*'), \text{mOp}(r, 3, '^'))$$

$$\left(\left(\left(\frac{4}{3} \right) * \pi \right) * (r^3) \right)$$

7. (10 points) Convert each of the following to the missing forms.

(a) (postfix:) 3 2 * 5 4 * +

prefix - + * 3 2 * 5 4

3

infix - 3 * 2 + 5 * 4

(b) (prefix:) * 3 - 2 + 4 5

infix - 3 * (2 - (4 + 5))

3

postfix - 3 2 4 5 + - *

(c) The volume of water in a round water balloon of radius r_b which contains three marbles of radius r_m .

infix - $\left(\left(4 * \pi \right) * \left(\left(\frac{r_b^3}{3} \right) - (r_m^3) \right) \right)$

prefix - $* * 4 \pi - \left(\frac{r_b^3}{3} - r_m^3 \right)$

postfix - $4 \pi * \left(\frac{r_b^3}{3} - r_m^3 \right)$

8. (10 points) Rewrite as Matlab expressions and add parentheses to show how Matlab would calculate the following expressions.

(a) $3 * 4 + 2 * 6 / 3 / 2^4 - 5^2$

$$\left\{ (3 * 4) + \left[\left(\left((2 * 6) / 3 \right) / (2 \wedge 4) \right) \right] - (5 \wedge 2) \right\}$$

(b) $3 > 8 || 4 < 2 + 5 * 6$

$$(3 > 8) || (4 < (2 + (5 * 6)))$$

(c) $3 > 6 | -5 == 2 | 1 < 7 \& 2 < 5$

$$\left\{ \left[(3 > 6) | ((-5) == 2) \right] | \left[(1 < 7) \& (2 < 5) \right] \right\}$$

9. (10 points) Consider the following logic expressions.

- (a) Check the expressions in which short circuiting is enabled.
- (b) Circle the expressions in which short circuiting actually occurs.
- (c) Evaluate each expression.
- (d) Where does a side effect of a function take place, and what happens?

Here are the expressions:

(a) $25 > 357 || true$ ✓ 1

(b) $-5 == 8 \&\& 21 > 9$ ✓ 0

(c) $45 < 24 | logical(fprintf('howdy'))$ 1

(d) $45 < 245 | logical(fprintf('howdy'))$ 1

(e) $45 < 245 || logical(fprintf('howdy'))$ ✓ 1

(f) $true || 2 > 71$ ✓ 1

(g) $2 - 16 < 6 \& 5 > 2$ 1

} side effect prints 'howdy'

10. (10 points) Write a piece of code that determines if t is above freezing, at freezing, or below freezing (degrees Kelvin), and prints out an appropriate message for each. Do this in each of the following ways.

- (a) Three separate *if* statements
- (b) One nested *if* statement
- (c) An *if* with *elseif*s

```
a) if t > 273
    above
end
if t == 273
    equal
end
if t < 273
    less
end
```

```
b) if t > 273
    above
else
    if t == 273
        equal
    else
        less
    end
end
```

```
c) if t > 273
    above
elseif t == 273
    equal
else
    less
end
```