**Answer the questions in the space provided. If you run out of space, continue on the back of the page.**

Name: A. Name

Date: 10/6/16 10+6=16

| Question: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Total |
|---|---|---|---|---|---|---|---|---|
| Points: | 3 | 5 | 4 | 5 | 10 | 10 | 8 | 45 |
| Score: | | | | | | | | |

**Multiple choice questions.**

For each of the following questions or statements, circle the letter correwsponding to one response that best answers the question or completes the statement.

1. (3 points) Which of the following is *not* a keyword in C?

   A. while

   B. if

   C. double

   D. print

   E. struct

2. (5 points) Which of the following commands will list files in the subdirectory called 'data' (no quotes in the name) of your current working directory?

   A. ls /data

   B. ls data

   C. ls ../data

   D. dir data

   E. dir /data

3. (4 points) Which of the following commands will transform the current directory into a Git repository?

   A. git init
   B. git init master
   C. git merge origin master
   D. git push origin master
   E. git remote add origin

4. (5 points) Which of the code fragments below will produce identical output?

1.

```
1  int x = 0, i;
2
3  for (i = 0; i < 1; i++)
4  {
5      printf ("%d\n", x);
6  }
```

prints one value

0

2.

```
1  int x = 0, i = 0;
2
3  while (i < 1)
4  {
5      printf ("%d\n", x);
6      i += 1;
7  }
```

prints one value

0

3.

```
1  int x = 0;
2
3  printf ("%d\n", x++);
```

First print, then increment

0

4.

```
1  int x = 0;
2
3  while (x < 1)
```

```
4       {
5           printf ("%d\n", ++x);
6       }
```

First increment, then print

A. only 1 and 2

B. only 3 and 4

C. only 1, 2, and 3

D. only 2, 3, and 4

E. all code fragments produce identical output

**Local and global variables**

5. Describe exactly but briefly what is the output of the program.

```
1  #include <stdio.h>
2
3  int fun (int x, int y, int z);
4
5  int main (void)
6  {
7      int x = 2, y = 2, z = 2;
8
9      printf ("main 1: %d  %d  %d\n", x, y, z);
10     z = fun (x, y, z);                          <- z=9
11     printf ("main 2: %d  %d  %d\n", x, y, z);
12
13     return (0);
14 }
15
16 int fun (int x, int y, int z)
17 {
18     x++;                    x=3
19     y += 1;                 y=3
20     ++z;                    z=3
21     printf ("fun:    %d  %d  %d\n", x, y, z);
22
23     return (x + y + z);     <- returns 9
24 }
```

local to fun variables, not visible outside of fun

(a) (1 point) The first printf statement in the main program:

main1: 2 2 2

(b) (3 points) The printf statement in the function fun:

fun: 3 3 3

(c) (6 points) The second printf statement in the main program:

main2: 2 2 9

**Arrays and pointers**

6. Describe exactly but briefly what is the output of the program.

```
1  #include <stdio.h>
2
3  #define ND  3
4
5  int fun (int *z);
6
7  int main (void)
8  {
9      int a[ND], z;
10
11     a[0] = 2, a[1] = 2, a[2] = 2;
12
13     printf ("main 1: %d  %d  %d\n", a[0], a[1], a[2]);
14     z = fun (a);
15     printf ("main 1: %d  %d  %d  %d\n", a[0], a[1], a[2], z);
16
17     return (0);
18 }
19
20 int fun (int *z)
21 {
22     (z[0])++;
23     z[1] += 1;
24     ++(z[2]);
25     printf ("fun:    %d  %d  %d\n", z[0], z[1], z[2]);
26
```

a[3] - array
z - just an integer, local to main

z = 9

should be "main2"

upon entrance z[0]=2, z[1]=2, z[2]=2

z[0]=3 → (a[0]=3)
z[1]=3 → (a[1]=3)
z[2]=3

z in fun is the same array as a in main

```
27      return (z[0] + z[1] + z[2]);
28  }
```

← returns 9

(a) (1 point) The first printf statement in the main program:

main1: 2 2 2

(b) (3 points) The printf statement in the function fun:

fun: 3 3 3

(c) (6 points) The second printf statement in the main program:

main2: 3 3 3 9

**Reading C code**

7. (8 points) Consider the code below.

```
1   #include <stdio.h>
2
3   int main (void)
4   {
5       int i, j;
6
7       for (i = 0; i < 4; i++)
8       {
9           for (j = 0; j < 3; j++)
10              printf ("*");
11          printf ("*\n");
12      }
13
14      return 0;
15  }
```

repeats four times

prints 3 asterisks on one line

adds 4th asterisks and goes to another line

How many asterisks does this code prints in total? 16

How many lines of asterisks is printed? 4