

NAME: Anthony Chavez Your Regular Section # 2

All problems count 3 points each unless a number in () indicates a different point count.
 Total point count for this test is 182 (20% of your grade).

→ **True and False Section. Circle the correct choice.**

1. The C language produces trig function values in degrees. 1. True False
2. UNIX became one of the earliest operating systems to be written in a high-level language, a fact that made subsequent porting to other hardware architectures possible. 2. True False
3. If a variable is declared inside a function or a block, it is available only to that function or block. 3. True False
4. Linux came before UNIX. 4. True False
5. The purpose of the **make utility** is to determine automatically which pieces of a large program need to be recompiled, and then issue the commands to recompile them. 5. True False
6. A **system call** is an explicit request to the kernel made via a **software interrupt**. 6. True False
7. A wrapper routine is in the Kernel Space rather than the User Space. 7. True False

→ **Multiple Choice Section. Pick the best answer.**

#8-9. (The **quotes** in these answers are **just to help** by marking the edges of the numbers so you can see the embedded spaces marked with a "b".)

8. What will get printed?
 double k = 6.789;
 printf("%4.2f", k);
 A. "6.789"
 B. "6.79"
 C. ".789"
 D. "b6.8"
 8. B
9. What will get printed?
 int i = 4, j = 9;
 printf("%d", i+j);
 A. "5"
 B. "13"
 C. "49"
 D. "4+9"
 9. B

10. The escape sequence to print a New Line is
A. n/ ☒ C. \n
B. /NL D. \N
10. C
11. Which of the following is the valid choice for a C program?
A. #include stdlib.h C. #include [stdlib.h]
B. #include "stdlib.h" ☒ D. #include <stdlib.h>
11. D
12. If a conversion specifier starts with a "-", it means:
A. right adjusted C. print in octal
☒ B. left adjusted D. print in hexadecimal
12. B
13. If a conversion specifier uses an "x", it means:
A. right adjusted C. print in octal
B. left adjusted ☒ D. print in hexadecimal
13. D
14. If a conversion specifier uses an "0", as in zero, it means:
A. right adjusted C. print in octal
☒ B. pad with zeros D. print in hexadecimal
14. B
15. Most code structures in C do not have a semi-colon at the end. Which of the following requires the use of an ending semi-colon.
A. do loop C. for loop
☒ B. do-while loop D. if-else-if
15. B
16. Which of the following are the operators used to form the conditional ternary operator that is a replacement for *if-else*?
A. ;; C. :?
☒ B. ?: D. %:
16. B
17. The term POSIX is an abbreviation for
A. Perennial Operating System Interface
B. Permanent Operating System Interface
C. Popular Operating System Interface
☒ D. Portable Operating System Interface
17. D
18. If the root is considered at the top, and you are in the lab4 directory, what will be accomplished by the Linux command: **cd ..**
A. move down one directory C. move to home base
☒ B. move up one directory D. move down two levels
18. B
19. The Linux command **pwd** stands for
A. password ☒ C. print name of working directory
B. portable word description D. print path of wandering direction
19. C

20. The standard directory named **/dev** holds information on
A. deviation files C. development files
☒ B. devices D. development folders 20. B
21. The command to display a simple list of the contents of the current directory in Linux is
☒ A. **ls** C. **cp**
B. **dr** D. **mylist** 21. A
22. The command to remove a file in Linux is
A. **ls** C. **rfile**
B. **rdir** ☒ D. **rm** 22. D
23. The command to copy a file in Linux is
A. **cat** ☒ C. **cp**
B. **cd** D. **copy** 23. C
24. What is the Linux command to find help on a command?
A. **clear** C. **help**
☒ B. **man** D. **manual** 24. B
25. Which of the following describes the action of the **chmod** command?
☒ A. sets permissions for files. C. searches files for strings
B. modifies the contents of a file. D. creates a file 25. A
26. File permissions are established for three categories. Which of the following is **not** one of those three groups?
A. **group** ☒ C. **permissions**
B. **other** D. **user** 26. C
27. Access permissions for files are **rwX**. The **r** stands for:
A. **record** C. **right**
☒ B. **read** D. **rich** 27. B
28. Which of the following is the correct name of the root directory?
A. **root** C. **/bin**
☒ B. **/** D. **/usr** 28. B
29. Which of the following is the command to create a new directory?
A. **create** C. **dir**
B. **cd** ☒ D. **mkdir** 29. D
30. When in **vim**, the correct way to exit from **insert** mode to **command** mode is
A. **Ctrl-d** ☒ C. **Escape key**
B. **F4** D. **Enter Key** 30. C

31. When in *vim*, the correct way to write/save changes and quit is
A. ctrl-d
B. :q
C. :q!
D. :wq
31. D
32. The four parts of program development are listed here alphabetically. Which one occurs **last** in the process?
a. compiler
b. editor
C. linker
D. preprocessor
32. C
33. The Linux debugger is called
A. gdb
B. gnu
C. debug
D. bug
33. A
34. The first thing we do in the debugger is set _____ that allow the program to stop at those points in the program.
A. points
B. stops
C. breaks
D. explore
34. C
35. In the debugger, what is the command to show the contents of a variable?
A. points
B. print
C. show
D. explore
35. B
36. The *scanf* and *fscanf* functions are very picky. When using one of these functions, and a variable declared as a type double, the required conversion specifier to use is:
A. %d
B. %i
C. %f
D. %lf
36. D

→ Fill in the answer as requested.

37. What is the result of the following C statement? Use the rules of precedence.
int a = 4, b = 1, c = 2, d = 3, e = 5;
int j;
j = a * (-b) + c * d;
4 * -1 + 2 * 3
-4 + 6 = 2
37. j = 2

38. (6) Circle **all** of the **valid** variable names (ANSI C).

area 2Area Area2
my_slope my-slope orig_radius

39. Write a line of code to *#define* the variable **JUMP** as the value of **15**.

```
#define JUMP 15
```

40. **Add** the appropriate symbol(s) to the words below so the line represents a *Comment* in ANSI C. Either solution is OK.

```
//Loop to read values
```

41. The code lists: **char a = 'M';**

Write a line of code to put that variable's contents on the screen using the function **putchar()**. Also use the variable **a**.

```
putchar(a);
```

42. Write a function prototype for a function named **sum_it** that will take as input two integer values **a** and **b** and return the sum as an integer.

```
int sum_it(int a, int b);
```

43. Rewrite the third line of code to force a non-integer division. Use casting.

```
1   int a, b;  
2   double c;  
3   c = a / b; → c = (double) a / (double) b;
```

44. (4) Complete the following code with a **printf** statement that will show the following on the screen: **The sum of 9 + 7 = 16**
Presume that the user typed in the value of **9** for **a** and the value **7** for **b**.

```
int a, b, c;  
printf("\nEnter two numbers: ");  
scanf("%i%i", &a, &b);  
c = a + b;  
printf("The sum of %i + %i = %i", a, b, c);
```

2

Problems 45-46-47 are related and should use the C language protocols.

45. (3) Declare a file pointer. (A short name is just fine.)

`FILE * p1;`

46. (3) Using your file pointer from #45, write a line of code to open a file named **f.dat** that will have **input** data. (NO error checking required)

`p1 = fopen("f.dat", "r");`

47. (3) Write ONE line of code to **read** data from the file you just opened above in #46. Use **double** variables **c** and **d**. No need for a loop of any sort include **while**.

`fscanf(p1, "%lf%lf", &c, &d);`

-
48. Write a Linux command to display, on the screen, the contents of a code file named **f3.c** (without using **vim** the editor).

`> cat f3.c`

49. Write a Linux command to show the **long** list of directory contents including the file permissions.

`> ls -l`

50. (4) Write a Linux command to compile a file named **lab14.c** that will have the executable named **area** instead of **a.out**. It will also need to use the **math** libraries.

`> gcc -o area lab14.c -lm`

→ more on next page →

Use this code for 51-52-53-54.

```

      position: 0 1 2 3 4 5
(line 1) int x[] = {4, 3, 6, 11, 7, 10}, *ptr = &x[2];
      ...
(line 2) *ptr = *ptr + 2; = 8 // {4, 3, 8, 11, 7, 10}
      6 + 2
(line 3) *(ptr-1) = x[5]; // {4, 10, 8, 11, 7, 10}
      x[1] = 10

```

51. What is the value of *ptr, after initialization, after line 1? 51. 6
52. What is the value of *ptr after the execution of line 2? 52. 8
53. What is the value of *(ptr-1) after the execution of line 3? 53. 10
54. (12) What are the values in the whole array when all three lines of code have been executed?

54. 4 10 8 11 7 10

55. (6) When the following code is finished executing, what will be the values in the array? Fill in the array/box below with the array values. (Rows and Columns are marked.)

```
int r, c, x[4][3];
```

```

for (r = 0; r < 4; r++)
{
    for (c = 0; c < 3; c++)

        x[r][c] = r - c;
}

```

	Column 0	Column 1	Column 2
Row 0	0	-1	-2
Row 1	1	0	-1
Row 2	2	1	0
Row 3	3	2	1