CS133 Chapter 13, 14, 16 Test <u>Mark Answers on Scantron Form</u>

1. Given the code fragment: `char s[20]; int i,j; scanf(“%d%s%d”, &i, s, &j);` if the user enters: `12abc34 56def78` at the keyboard, what will the values of i, j, and s be?

a) i=12, s=abc, j=34 **b) i=12, s=abc34, j=56** c) Nothing. The program will fail. d) none of the above

2. How many bytes are stored for the string literal `“CS133"`

a) 5 **b) 6** c) 7 d) 8

3. How many bytes are stored for the string literal `“CS133\n”`

a) 5 b) 6 **c)** 7 d) 8

4. Which of the following `printf` statements will compile without a syntax error

a) `printf(‘Hello World\n’);` b) `printf(‘Hello World’);`

c) `printf(“”);` d) `printf(“Hello World\n”);` **e) (c) and (d)**

5. Which of the following is true?

a) All squares are rectangles b) All strings are char arrays c) all char arrays are strings d) all of the above **e) (a) and (b)**

6. Given: `char a[10], b[10]=”CS133";` which of the following will copy `“CS133"` into character array **a**

a) `a = b;` b) `a = “CS133"` c) `strcat(b,a);` d) `strcpy(b,a);` **e) `strcpy(a,b);`**

7. Given: `char a[]=“CS133\n”;` what is the value of `a[6]`

a) `‘0'` b) `“0"` **c) `0`** d) `‘n’` e) `92`

8. Given the statement `len = strlen(“CS133\n”);` What is the value of `len`

a) 5 **b) 6** c) 7 d) 8 e) none of the above

9. Given the code fragment:

```
char a[40],b[]=”Oregon”,c[]=”is a blue state.”; strcpy(a,b);
xxxxxx(a,” “);
strcat(a,c);
printf(a);
```

Which of the following should be substituted for `xxxxxx` to print `“Oregon is a blue state.”` on the monitor

a) `strcmp` b) `strcat` **c) `strcpy`** d) `strlen` e) none of the above

10. Given the code fragment: `char str[]=”CS133\n”; xxxxx; printf(“%s”,str);` When statement `xxxx` is replaced any one of three of the following statements the same printout occurs. Which one of the four is not equivalent to the other three

a) `str[0] = 0;` b) `str[0] = ‘\0';` c) `strcpy(str, “”);` **d) `strcat(str, “”);`**

11. What gets printed to the screen, given the (confusing) code fragment:

```
#define PI 3.14);
double d = 2.0;
printf(“When d = %.1f, c = %.2f\n”, d, d*PI
```

a) Nothing, the code won’t compile **b) `When d = 2.0, c = 6.28`**

12. If a programmer wishes to create the header file **`aheader.h`** for their own use, which of the following **`include`** statements would most likely be placed at the beginning of the source file(s)

a) **`include <aheader.h>`** b) **`include “aheader.h”`**

c) **`#include <aheader.h>`** **d) `#include “aheader.h”`** e) **`#define aheader.h`**

13. If a programmer wanted to reference the standard **`stdio.h`** header, which of the following **`include`** statements could be used

a) **`#include <stdio.h>`** b) **`#include “stdio.h”`** c) **`include<stdio.h>`**

d) all of the above **e) both (a) and (b)**

14. According to the class lectures, the most common use of **`include`** files is to share

a) function and struct declarations b) function definitions (executable code) c) defined symbols

d) machine code e) (a) and (c)

15. Including the same header file more than once in a source file is always an error

a) true **b) false**

16. Assuming that **`string1 = “hello”`** and **`string2 = “hello world”`**, Which of the following returns **0**?

(a) **`strcmp(string1, string2);`** (b) **`strcmp(string1, string2, 6);`**

(c) **`strncmp(string1, string2, 6);`** **(d) `strncmp(string1, string2, 5);`**

17. When constructing a header file, why it is common to use the following sequence of preprocessor directives (UNIQUE_SYMBOL probably is derived from the header file name):

```
#ifndef UNIQUE_SYMBOL
#define UNIQUE_SYMBOL
.. the body of the header file...
#endif
```

a) to establish the code’s copyright **b) to allow the same header file to be included more than once**

c) it’s a useless tradition from IBM d) to speed up the compilation process e) none of the above

18. The proper format for this **`struct`** is

```
(a) struct Time      (b) struct Time {      (c) struct Time {     (d)  struct Time
      int minute            int minute,            int minute;    {
      int hour;             int hour,              int hour;           int
                         }                      }                 minute;
                                                                       int hour;
                                                                  };
```

19. By default, structures are passed to functions

(a) pass-by-value, one byte at a time (b) pass-by-value, one member at a time

(c) structures cannot be passed between functions (d) by inversion

20. How many bytes are stored for the string literal **`“CS133/n”`**

a) 5 b) 6 **c) 7** d) 8

21. Given the code fragment:

```
typedef struct {short int x,y;}point;
typedef struct {point ctr; double radius;}circle;
circle c1,c2={{0,0},4.9};
c1 = c2;
```

How would you move the circle object **c1** 2 units to the right of **c2**

a) `c1 = c2 + 2;` b) `c1.x = c2.x + 2;` c) `c1.ctr.x += 2;` d) none of the above

22. Given the code fragment:

```
typedef struct {short int x,y;}point;
typedef struct {point ctr; double rad;}circle;
circle c1={{0,0},4.9},c2={{0,0},4.9};
```

How would you correctly compare c1 and c2 for identity?

a) `if(c1 = c2)` b) `if(c1 = = c2)`

c) `if(c1.x = = c2.x && c1.y = = c2.y && c1.rad = = c2.rad)`

d) `if(c1.ctr = = c2.ctr && c1.rad = = c2.rad)`

e) `if(c1.ctr.x = = c2.ctr.x && c1.ctr.y = = c2.ctr.y && c1.rad = = c2.rad)`

23. Given the code fragment:

```
typedef struct {short int x,y;}point;
typedef struct {point ctr; double rad;}circle;
typedef union {point ctr; double rad;}example;
short int sizeCircle = sizeof(circle),sizeExample = sizeof(example);
```

a) `sizeCircle = = sizeExample` b) `sizeCircle < sizeExample`

c) `sizeCircle > sizeExample`

24. Given the code fragment:

```
typedef enum {black, brown, red, orange, yellow, green, blue,
     violet, grey, white} ColorCode;
ColorCode band1=brown, band2=yellow, band3=red;
if(band1 = = brown) printf(“band two = \”%d \”“, band2);
```

What does printf display on the monitor

a) “yellow” b) 4 c) “4" d) none of the above

25. What is the final value of p given the defined macro

```
#define mystery(a,b) ((a)>(b))?(a):(b)
```

and the code fragment

```
int m=5, n=10, p=15;
p = mystery(m,n);
```

a) 5 **b) 10** c) 15 d) 0

26. The **null character** terminates all strings. What is the **null character** ?

a) a byte of all zero bits b) “0" c) ‘0' d) ‘\0' **e) both (a) and (d)**