Key for practice final exam

1. B
2. B
3. D
4. A
5. C
6. A
7. B
8. A
9. C
10. A
11. D
12. C
13. A
14. A
15. C
16. C
17. B
18. D
19. C
20. D
21. C
22. A
23. B
24. A
25. D
26. B
27. D
28. A
29. B
30. A
31. Ensures that the program can use the monitor, keyboard, mouse and other I/O devices
32. A

```
#Include <stdio.h>
int main (){
        int first, second;
        printf(“please enter the first number \n”);
        scanf (“%d”, &first);
        printf(“please enter the second number \n”);
        scanf(“%d”, &second);
        printf(“%d times %d equals %d \n”, first, second, first*second);
```

```
 return 0;
}
```

33. 1
34. An array name is just a pointer to the beginning of the array in memory. So when you print the array name, you get a pointer - a memory address. By contrast, array elements are variables of specific types - in this case integers. So when you print the array elements, you are printing integers.
35. Compiling translates high level languages, such as C, to machine code. High level code in, machine code out; nothing executes. Executing code causes machine code to be run on the computer, producing answers.
36. Dividing 9 by 4.500 yields an answer of 2.0
37.

```
for (i = 1; i <=10; i++){
        for (j = 1; j <= 10; j++) {
                printf(“*”);
        }
        printf (“\n”):
}
```

38.

```
 for (i = 1; i <=10; i++){
        for (j = 1; j <= 10; j++) {
                printf(“%d  ”, i*10 + j);
        }
        printf (“\n”):
}
```

39.

```
for (i = 1; i <= 10; i++) {
        printf(“%d ”, i)
}
```

40. w = 9 x = -74 y = 80 z = 1

41. You have a while loop that checks input to make sure a valid value has been entered. If the user enters a valid value the first time, the while loop makes sure that the body of the loop is never executed.

42.

```
 while (answer != “q”){
      printf("Please enter one letter. Enter q to quit the program.\n");
      scanf ("%c%c", &answer, &cr);


}
```

43.

```
  switch(x) {
      case (1): printf("Excellent");break;
      case (2): printf("Excellent"); break;
      case (3): printf(“Good”): break;
      case (4): printf(“Good”); break;
      default: printf(“Need to do better”);
      }
```

44. The statement

```
scanf ("%d", num)
```

```
    Tries to write the value into the symbol table directly.  You have to say
            Scanf (“%d”, &num);
    To go to the location in memory pointed to by the symbol table, and write
the value to the location in memory.
```

```
45.   int summarize(int nums){
          int total = 0;
          for (i = 1; i <= nums; i++) {
               total += i;
          }
          return total;
      }
```

46. True
47. False
48. True
49. False
50. True
51. False
52. True
53. False
54. True
55. False
56. False
57. True
58. True
59. True
60. False