Assignment 4 (with Solutions)

1. The control/conditional statements used in C is/are
   a) if-else statements
   b) switch statements
   c) Both (a) and (b)
   d) None of these

   Solution: (c) Both if-else and switch statements are conditional statements in C.

2. What is the other statement that can avoid multiple nested if conditions?
   a) Functions
   b) Switch statements
   c) If-else statements with 'break'
   d) Loop statements

   Solution: (b) switch statements are used to avoid multiple nested if conditions.

3. In the following example, tell which statement is correct?

```
    if ( (Condition1==1) && (Condition2==1))
      printf("hallelujah");
```

   a) Condition1 will be evaluated first, Condition2 will be evaluated second
   b) Condition2 will be evaluated first, Condition1 will be evaluated second
   c) Condition1 will be evaluated first, Condition2 will be evaluated only if Condition1 is TRUE
   d) Condition2 will be evaluated first, Condition1 will be evaluated only if Condition2 is TRUE

   Solution: (c) Condition1 will be evaluated first; Condition2 will be evaluated only if Condition1 is TRUE. This is called the Short-circuited evaluation of the operators

4. What will be the output?

```
#include <stdio.h>
int main()
  {
     int x = -0.5;
     if (!x)
        printf("hello");
  return 0;
  }
```

   a) Hello
   b) Compilation error
   c) Nothing will be displayed
   d) Compiler dependent

   Solution: (a) hello

5. Which statement is correct?
   a) Switch case can only be used with 'int'
   b) Switch case can only be used with 'char'
   c) Switch case can be used with all data type

d) Switch case can only be used with ‘int’ and ‘char’

Solution: (d) Switch case can only be used with ‘int’ and ‘char’. Floats or other data types are not allowed

6. What is the output of the following C code?

```
#include <stdio.h>
int main()
  {
   int x = 0;
   if (x == 1)
       if (x >= 0)
           printf("true\n");
       else
           printf("false\n");
    return 0;
    }
```

a) true
b) false
c) Depends on computer
d) Nothing is printed.

Solution: (d) x is initialize with 0 and the if statement compares it with 1, thus the if condition is false and the nested if statements do not get executed. Hence, the program do not prints anything.

7. Which of the following statement is equivalent to

```
x= a++ * b;
```

a) a=a+1;
   x=x*b;
b) x=a*b;
   a=a+1;
c) x=a*b;
   x=x+1;
d) None

Solution: (b)

8. What is the following statement equivalent to

```
x= --a * b;
```

a) a=a-1;
   x=a*b;
b) x=a*b;
   a=a-1;
c) x=a*b;
d) None

Solution: (a)

9. What will be the values of a, b, c after the execution of the followings
   ```
   int a=2, b=3, c=0;
   c += --a * b++;
   ```
   a) a=3, b=9, c=4
   b) a=1, b=3, c=6
   c) a=1, b=4, c=6
   d) a=1, b=4, c=3

   Solution: (d) c = c + (--a * b++)

10. What will be the values of a, b, c after the execution of the following
   ```
   int a=20, b=5, c=2;
    c /= ++a * --b;
   ```
   a) a=21, b=4,c=0;
   b) a=20, b=4,c=2;
   c) a=20, b=5,c=0;
   d) a=21, b=5,c=0;

   Solution: (a)

11. What will be the printed value of digit of the following program?
   ```
   #include <stdio.h>
   main()
   {
      int digit=0;
      while (digit>=9)
         digit++;
      printf("%d", digit);
   }
   ```
   a) 1
   b) 0
   c) 10
   d) 9

   Solution: (b) The while condition is false. Thus the statement inside loop will never be executed. Hence the printed value of digit is 0.

12. The loop which is executed at least one is
   a) while
   b) do-while
   c) for
   d) none of the above

   Solution: (b) do-while loop is executed at least one even though the condition is false.

13. while(1) is used in a program to create
   a) False statement
   b) Infinite loop

c) Terminating the loop
d) Never executed loop

Solution: (b) while(1) is used to create infinite loop.

14. Is there anything wrong in the following program segment?

```
switch(someCharacter)
{
case ‘a’:
case ‘e’:
case ‘i’:
case ‘o’:
case ‘u’:
printf(“Vowel”);
break;
default:
printf(“Consonant”);
}
```

a) Yes- case blocks are not defined
b) No- it’s all O.K.
c) Compiler Dependent
d) More info required

Solution: (b) It’s all fine. Unlike with for loops, or ifelse blocks, multiple statements can follow each other without curly braces

15. What is wrong with the following program segment?

```
switch(temperature < 15 && windy == TRUE)
{
case !windy:
        printf(“cold day”);
        break;
default:
    printf(“hot day”);
}
```

a) Nothing is wrong
b) Variable cannot be used in individual cases
c) Will print both the statements
d) Break is not allowed when using variable in a case checking

Solution: (b) the switch portion can contain any constant or expression, but the individual case conditions can only include constants, or expressions of constants (no variables allowed).