Week 5: Assignment Solution 2023

1. "continue" statement in C is used
    a) to continue to the next line of code
    b) to debug
    c) to stop the current iteration and begin the next iteration from the beginning
    d) None of the above are true

Solution: (c)

2. The operators << and >> are
    a) Relational operator
    b) Logical operator
    c) Assignment operator
    d) Bitwise shift operator

Solution: (d) Bitwise shift operator

3. Compute the printed value of i of the C program given below

```
#include <stdio.h>
int main()
{
   int i = 0, j = 0;
   while (i < 4, j < 5)
   {
      i++;
      j++;
   }
   printf("%d, %d\n", i, j);
   return 0;
}
```

a) 4,5
b) 4,4
c) 5,5
d) 0,0

Solution: (c) The while condition checks the last condition (i.e. j<5) and till the condition is satisfied the block inside the loop is executed. Thus the loop is run for 5 times and both the values of i and j are incremented by 5.

4. How many times 'Hello' will be printed while executing the below C code?

```
#include <stdio.h>
int main()
{
   int i = 0;
   int j = 0;
   for (i = 0; i < 4; i++)
   {
      for (j = 0; j < 5; j++)
      {
         if (i > 0)
            continue;
            printf("Hello \n");
      }
   }
return 0;
}
```

a) 4 times
b) 20 times
c) 5 times
d) no print

Solution: (c) Only in the first iteration i=0 and the if condition is not satisfied, thus for j=0 to 5 i.e. 5 times Hello will be printed. From the next iteration of the first for loop, i will be 1 and the if condition becomes true and thus the loop continues without executing the printf statement. Hence, only 5 times Hello will be printed.

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

```
#include <stdio.h>
int main()
{
   int a = 1;
   if (a--)
      printf("True\n");
   if (++a)
      printf("False\n");
    return 0;
}
```

a) True
b) False
c) Both 'True' and 'False'
d) Compilation error

Solution: (c) 'a--' post-increment the value of a. Thus, the if statement is executed as the value of a is considered as 1 which is true. '++a' pre-increment the value of a. Thus, the decremented value of a (which is 0) is incremented first and then assigned. So, both the if statements are executed ad correspondingly both True and False will be printed.

6. Find the output of the following C program

```
#include <stdio.h>
int main()
{
   int i = 0;
   if(i==0)
   {
            i=i+1;
            break;
   }
   printf("%d", i);
   return 0;
}
```

a) 0
b) 1
c) No output
d) Compiler error

Solution: (d) Break statement is applicable in loop and switch statements. It is not allowed inside if statement. Thus the program will show compiler error.

7. How many times the 'Hello' will be printed in the below C code?

```
#include <stdio.h>
int main()
{
   int k = 0;
   for (;; k++)
   {
         printf("Hello");
         if(k%10==0)
         break;
   }
 return 0;
}
```

a) 1 time
b) 10 times
c) 11 times
d) Compilation error

Solution: (a) As the initial value of k is 0, the if condition is satisfied and the break statement is executed. Thus the control of the program comes out of the loop and only once Hello is printed.

8. How many times Hello will be printed in the below C code?

```
#include <stdio.h>
int main()
{
   int k, j;
   for (k=0; k<=10; k+=2)
   {
            for(j=1; j!=k; j=j+1)
            {
                     printf("Hello \n");
                     break;
            }
   }
return 0;
}
```

a) 10 times
b) 5 times
c) 6 times
d) Infinite times

Solution: (c) for j=1, the inner for loop is always executed. Thus, it will print Hello and come out of the inner loop. Thus, hello will be printed six times (i.e. for k=0,2,4,6,8 and 10).

9. What will be the output?

```
#include <stdio.h>
int main()
{
   int x;
   x = 4 < 8 ? 5 != 1 < 5 == 0 ? 1: 2: 3;
   printf("%d", x);
   return 0;
}
```

a) 1
b) 2
c) 3
d) Error

Solution: (b) 2
exp1? exp2: exp3
4 > 8 ? 5 != 1 < 5 == 0 ? 1 : 2 : 3;

exp1 is True, so exp2 will be evaluated, which is false (1<0). So, 2 will be printed.

10. What is the output of the following C program?

```
#include <stdio.h>
int main()
{
   int a = 0, i = 0, b;
   for (i = 0; i < 5; i+=0.5)
   {
      a++;
      continue;
   }
   printf("%d", a);
   return 0;
}
```

a) 5
b) 10
c) No output
d) Compilation error

Solution: (c) As i is initialized as an integer variable, integer value of i after the operation (i=i+0.5) will be zero. Thus, the loop will never be ended and the control will not come to the printf statement at all. So, nothing will be printed.