ASSIGNMENT – 5 SOLUTION

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

Solution: (c)

2. In C three way transfer of control is possible using
    a) Unary operator
    b) Logical operator
    c) Ternary operator
    d) None

Solution: (c) Ternary operator

3. What will the following C code prints?

```
#include <stdio.h>
int main()
{
  int i=0;
  do
  {
    printf("while vs do-while\n");
  }while(i==0);
  printf("Out of loop");
   return 0;
}
```

    a) ‘while vs do-while’ once
    b) ‘Out of loop’ infinite times
    c) Both ‘while vs do-while’ and ‘Out of loop’ once
    d) ‘while vs do-while’ infinite times

Solution: (d) As the condition inside the while statement is always true, the loop will be executed infinite times and the statement inside the loop will be printed infinite number of times.

4. Compute the printed value of i and j 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);
}
```

    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.

5. 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.

6. 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;
}
```

Solution: 5 times

Solution: 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.

7. What will be printed on executing 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.

8. 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.

9. How many times the ‘Hello’ will be printed on executing the below C code?

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

Solution: 1 time

Solution: 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.

10. How many times “Hello” will be printed on executing 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;
}
```

Solution: 6 times. 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).

11. What is the output of the below C program?

```
#include <stdio.h>
int main()
{
   short int k=1, j=1;
   while (k <= 4 || j <= 3)
   {
            k=k+2;
            j+=1;
   }
   printf("%d, %d", k, j);
return 0;
}
```

a) 5, 4
b) 7, 4
c) 5, 6
d) 6, 4

Solution: (b) The loop will be continued till any of the condition k<=4 or j<=3 is satisfied. So, the loop will be executed 3 times. Thus, the value of k and j would be 7 and 4.

12. What will be the output?

```
#include <stdio.h>
int main()
 {
  char x=0;
  for(;x++;printf("%d", x)) ;
  printf("%d", x);
  return 0;
}
```

a) 0 1 2 ….infinite times
b) 0
c) 1
d) Compilation error

Solution: (c) 1
The semicolon at the end of for loop makes the for loop to run only one time making the value of x to be increased by 1, making x=1. Thus 1 is printed.

13. What will be the output on executing the following programme?

```
#include <stdio.h>
int main()
{
   int i=0;
   for(;;)
   {
      if(i==10)
         continue;
      printf("%d ",++i);
   }
   return 0;
}
```

a) 0 1 2 3 4 5 6 7 8 9 10 then it goes to infinite times
b) 1 2 3 4 5 6 7 8 9 10 then it goes to infinite times
c) Nothing will be printed
d) Compiler error

Solution: (b) 1 2 3 4 5 6 7 8 9 10 then it goes to infinite times
After printing 1 to 10, due to the “continue” statement, the printf statement will always be skipped and loop will iterate without printing anything.

14. What will be the output on executing the following C code?

```
#include <stdio.h>
int main()
 {
   int x=1;
   do
    {
```

```
        ++ x;
         continue;
         printf("%d" ,x);
         break;
      }while(x<=2);
         printf("After loop x=%d",x);
    printf("\n");
    return 0;
}
```

a) 12After loop x=2
b) 23After loop x=3
c) After loop x=3
d) No output

Solution: (c) After loop x=3

Initially the value of x will be 1 and it inters into the do-while loop where the value of x becomes 2 due to the pre-increment operator (++x). However due to "continue" statement, the lines after the "continue" statement are skipped and the control again goes to the statement ++x (as x<=2) where its value becomes 3 and again due to "continue" statement the rest of the lines are not executed and it finally comes out of the loop and "After loop x = 3" is printed.

15. 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) Compilation error

Solution: (c) 3
exp1? exp2: exp3
4 > 8 ? (5 != 1 < 5 == 0 ? 1 : 2) : 3;
exp1 is false, so exp2 will not be evaluated. 3 will be printed.