

## Week 5 Assignment Solution

1. The statement that transfers control to the beginning of the loop is called
  - a) break
  - b) continue
  - c) goto
  - d) None of the above

Solution: (b) continue

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 is the output of the following code?

```
#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. What is the output of the following C program?

```
#include <stdio.h>
int main()
{
    int a = 0, i;
    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

## Week 5 Assignment Solution

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.

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 and correspondingly both True and False will be printed.

6. What will be the output?

```
#include <stdio.h>
int main()
{
    int x=1;
    do
    {
        continue;
        printf("%d", x);
        x++;
        break;
    } while(x<=10);
    printf("\nAfter loop x=%d", x);
    printf("\n");
    return 0;
}
```

- a) After loop x=1
- b) 1  
After loop x=2
- c) 1 2 3 4 5 6 7 8 9 10
- d) No output

Solution: (d) No output

do while is an exit controlled loop, here loop body executed first, then condition will be checked. However due to continue statement, the lines after the continue statement are skipped. Hence nothing will be printed.

## Week 5 Assignment Solution

7. What will be the output?

```
#include <stdio.h>
int main()
{
    float k = 0;
    for (k = 0.5; k < 3; k++)
        printf("I love C\n");
    return 0;
}
```

- a) Error
- b) I love C - will be printed 3 times
- c) I love C - will be printed 6 times
- d) I love C - will be printed 5 times

Solution: (b) I love C will be printed 3 times

8. 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: (c) 3

exp1? exp2: exp3

(4 < 8) ? (5 != 1 < 5 == 0) ? 1 : 2 : 3;

exp1 is true, so exp2 will be evaluated, which is true so 2 will be printed.

9. The following program is used to find the reverse of a number using C language. Find the missing condition inside while statement (indicated as 'xxxx').

```
#include <stdio.h>
int main()
{
    int n, reversedNumber = 0, remainder;

    printf("Enter an integer: ");
    scanf("%d", &n);

    while(xxxx)
```

## Week 5 Assignment Solution

```
{
    remainder = n%10;
    reversedNumber = reversedNumber*10 + remainder;
    n /= 10;
}

printf("Reversed Number = %d", reversedNumber);

return 0;
}
```

- a)  $n \neq 0$
- b)  $n == 0$
- c)  $n \% 10 == 0$
- d)  $n / 10 == 0$

Solution: (a) The loop should be continued till the value of n becomes zero. Thus, the right option is  $n \neq 0$ .

10. Compute the printed value of i & j of the C program given below

```
#include <stdio.h>
int main()
{
    int i = 0, j = 15;
    while (i < 8, j > 9)
    {
        i++;
        j--;
    }
    printf("%d, %d\n", i, j);
    return 0;
}
```

- a) 8,10
- b) 8,9
- c) 6, 9
- d) 7, 10

Solution: (c) The while condition checks the last condition (i.e.  $j > 9$ ) and till the condition is satisfied the block inside the loop is executed. Thus the loop is run for 6 times. i will be incremented by 6 and j will be decremented by 6. The final values of i and j will be  $i=6$  and  $j=9$ .