

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

Yes, the answer is correct.

Score: 1

Accepted Answers:

b) continue

2) In C three way transfer of control is possible using

- a) Unary operator
- b) Logical operator
- c) Ternary operator
- d) None

Yes, the answer is correct.

Score: 1

Accepted Answers:

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

Yes, the answer is correct.

Score: 1

Accepted Answers:

d) 'while vs do-while' infinite 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

Yes, the answer is correct.

Score: 1

Accepted Answers:

c) No output

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

Yes, the answer is correct.

Score: 1

Accepted Answers:

c) Both 'True' and 'False'

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

Yes, the answer is correct.

Score: 1

Accepted Answers:

d) No output

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

No, the answer is incorrect.

Score: 0

Accepted Answers:

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

No, the answer is incorrect.

Score: 0

Accepted Answers:

b) 2

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)
    {
        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) $n / 10 == 0$

Yes, the answer is correct.

Score: 1

Accepted Answers:

a) $n \neq 0$

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

Yes, the answer is correct.

Score: 1

Accepted Answers:

c) 6, 9

Week 5 : Programming Assignment 1

Write a C program to check whether a given number (N) is a perfect number or not.
[Perfect Number - A perfect number is a positive integer number which is equals to the sum of its proper positive divisors. For example 6 is a perfect number because its proper divisors are 1, 2, 3 and it's sum is equals to 6.]

Sample Test Cases

	Input	Output
Test Case 1	8128	8128 is a perfect number.
Test Case 2	8000	8000 is not a perfect number.
Test Case 3	6	6 is a perfect number.
Test Case 4	87	87 is not a perfect number.

```
#include <stdio.h>
int main()
{
    int N;
    scanf("%d",&N); /* An integer number taken as input from test cases */

    int i, sum=0;
    for(i=1; i<N;i++)
    {
        if(N%i==0)
            sum+=i;
    }

    if(sum==N)
        printf("\n%d is a perfect number.",N);
    else
        printf("\n%d is not a perfect number.",N);
}
```

Week 5 : Programming Assignment 2

Write a C program to count total number of digits of an Integer number (N).

Sample Test Cases

	Input	Output
Test Case 1	45667	The number 45667 contains 5 digits.
Test Case 2	87	The number 87 contains 2 digits.
Test Case 3	3008	The number 3008 contains 4 digits.
Test Case 4	123456	The number 123456 contains 6 digits.

```
#include <stdio.h>
int main()
{
    int N;
    scanf("%d",&N); /*The number is accepted from the test case data*/
```

```
int temp, count;
count=0;
temp=N;
while(temp>0)
{
    count++;
    temp/=10;
}
printf("The number %d contains %d digits.",N,count);
}
```

Week 5 : Programming Assignment 3

Write a C program to check whether the given number(N) can be expressed as Power of Two (2) or not.

For example 8 can be expressed as 2^3 .

Sample Test Cases

	Input	Output
Test Case 1	256	256 is a number that can be expressed as power of 2.
Test Case 2	800	800 cannot be expressed as power of 2.
Test Case 3	8	8 is a number that can be expressed as power of 2.
Test Case 4	66	66 cannot be expressed as power of 2.

```
#include <stdio.h>
int main()
{
    int N;
    scanf("%d",&N); /* The value of N is taken from the test case data */

    int temp, flag;
    temp=N;
    flag=0;

    while(temp!=1)
    {
        if(temp%2!=0){
            flag=1;
            break;
        }
        temp=temp/2;
    }

    if(flag==0)
        printf("%d is a number that can be expressed as power of 2.",N);
    else
        printf("%d cannot be expressed as power of 2.",N);
}
```


Week 5 : Programming Assignment 4

Write a C program to find sum of following series where the value of N is taken as input

$$1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots 1/N$$

Sample Test Cases

	Input	Output
Test Case 1	100	Sum of the series is: 5.19
Test Case 2	20	Sum of the series is: 3.60
Test Case 3	6	Sum of the series is: 2.45
Test Case 4	50	Sum of the series is: 4.50

```
#include<stdio.h>
int main()
{
    int N;
    float sum = 0.0;
    scanf("%d",&N); /*Read the value of N from test cases provided*/

    int i;
    for(i=1;i<=N;i++)
        sum = sum + ((float)1/(float)i);
    printf("Sum of the series is: %.2f\n",sum);
}
```