STANDARD 10

COMPUTER SCIENCE

UNIT 01

CHAPTER 05

LOOP CONTROL STRUCTURES IN C

1. What is a loop/iteration?

A: A loop is a sequence of statements that is executed repeatedly to obtain a desire result or to meet a predetermined condition.

2. What is the while statement? Write the general form of while statement.

A: The 'while' is a loop statement that executes a set of instructions based upon a conditional test.
Syntax:

```
while(condition)
{
    loop body
}
```

3. What is the do while statement? Write the syntax of do-while statement.

A: The do-while is a loop statement that executes a set of statements as long as a specified condition is satisfied/true.
Syntax:

```
do
{
    loop body
}while(condition);
```

4. What is the for statement? Write the syntax of for statement.

A: The 'for' statement is a loop statement generally used to execute a block of statements a specified number of times depending on a condition. It is also known as count-controlled loop statement.
Syntax:

```
for(<initial>; <condition>; <increment>)
{
    loop body
}
```

5. What is the main difference between while() loop and do-while() loop?

A: Difference between while() loop and do-while loop are

while loop	do-while loop
1. The 'while' is a loop statement that executes a set of instructions based upon a conditional test.	1. The do-while is a loop statement that executes a set of statements as long as a specified condition is satisfied/true.
2. The statement block will not be executed when the value of the condition is false.	2. The statement block will not be executed when the value of the condition is false, but the block is executed at least once irrespective of the value of the condition.

6. What is an infinite loop? Give two examples.

A: An infinite loop is a sequence of instructions in a computer program which loops endlessly, either due to the loop having no terminating condition, having a condition that can never be met, or a condition that causes the loop to start over.
E.g., for(;;);, while(1){};, etc.

7. Which loop is known as count-controlled loop/counter type loop?

A: The 'for' loop is known as count-controlled loop.

8. What is the 'goto' statement?

A: The goto statement is an unconditional transfer of control statement. It is used to transfer the control from one part of the program to another. The place to which the control is transferred is identified by a statement 'label'.
Syntax: goto label;

9. What is the break statement?

A: The break statement is used to transfer the control to the end of a statement block in a loop.

10. What is the continue statement?

A: The continue statement is used to transfer the control of the beginning of a statement block in a loop.

11. What is the difference between break and continue statements?

A: The 'break' statement passes the control to the statement following the last statement of the closest iteration or the closest switch statement-block whereas the 'continue' statement skips the following statements within the loop and the control is transferred to the loop controlling expression of the iteration it belongs to.

12. What is a nested loop?

A: A loop embedded within another loop is called nested loop.

13. Explain Counter-controlled loop and Sentinel-controlled loop with examples.

A: Counter-controlled loops: Counter-controlled loop is a loop in which the exact number of iterations is known before the loop starts executing.
Example:

```
for(i=1;i<=10;i++)
    printf("*");
```

Sentinel-controlled loops: A sentinel-controlled loop is an indefinite repetition loop in which the number of iterations is not known before the loop starts executing.
Example:

```
scanf("%d",&x);
while(x!=0)
{
    printf("\n%f ",sqrt(x));
}
```

Rewrite the following program segment using while statement and also give the output.

```
int i, j;
for(i=1; i<=10; i++)
{
    for(j=1; j<=i; j++)
        printf(%3d');
    printf("\n");
}
```

A: Using while statement:

```
int i, j;
i = 1;
while (i <= 10)
{
    j = 1;
    while (j <= i)
    {
        printf("%3d ", j);
        j++;
    }
    printf("\n");
    i++;
}
```

Using do-while statement:

```
int i, j;
i = 1;
do
{
    j = 1;
    do
    {
        printf("%3d ", j);
        j++;
    }while (j <= i);
    printf("\n");
    i++;
}while (i <= 10);
```

14. Write a C program to find the sum of first n natural numbers.

```
#include<stdio.h>
#include<conio.h>
main()
{
    int n, i, sum = 0;
    printf("Enter the number\n");
    scanf("%d", &n);
    for(i = 1; i <= n; i++)
    sum = sum + i;
    printf("Sum = %d", sum);
    getch();
    return 0;
}
```

15. Write a C program to find the GCD/HCF of two integers.

A:
```
#include<stdio.h>
#include<conio.h>
int main()
{
    int n1, n2, i, gcd;
    printf("Enter two integers: ");
    scanf("%d %d", &n1, &n2);
    for(i=1; i <= n1 && i <= n2; ++i)
    {
        if(n1%i==0 && n2%i==0)
        gcd = i;
    }
    printf("G.C.D of %d and %d is %d",
    n1, n2, gcd);
    return 0;
}
```

16. Write a C program to check whether a number is Palindrome or not.

A:
```
int main()
{
    int n, reversed = 0, remainder, origi
    nal;
    printf("Enter an integer: ");
    scanf("%d", &n);
    original = n;
    while (n != 0)
    {
        remainder = n % 10;
        reversed = reversed * 10 + re
        mainder;
        n /= 10;
    }
    if (original == reversed)
        printf("%d is a palindrome.",
        original);
    else
        printf("%d is not a palindrome.",
        original);
    return 0;
}
```

17. Write a program to find the value of the power of a number m raised by an integer n. (without pow function)

A:
```
#include <stdio.h>
int main(void)
{
    int m, n;
    long double result = 1.0;
    printf("Enter a base number: ");
    scanf("%d", &m);
    printf("Enter an exponent: ");
    scanf("%d", &n);
    while (n != 0)
    {
        result *= m;
        --n;
    }
    printf("Answer = %.0Lf", result);
    return 0;
}
```

18. Write a C program to print a triangle in the following pyramid:

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

A:
```
#include <stdio.h>
int main()
{
    int n = 5;
    int i, j;
    for (i = 1; i <= n; i++)
    {
        // Print leading spaces
        for(j = 1; j <= n - i; j++)
        {
```

```
            printf("  ");
        // Two spaces for formatting
        }
        // Print left side of the pattern
        for (j = 1; j <= i; j++)
        {
            printf("%d ", j);
        }
        // Print right side of the pattern
        (ex cluding the center)
        for (j = i - 1; j >= 1; j--)
        {
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}
```

19. Write C program to print all the prime numbers between two given numbers.

A:
```
#include <stdio.h>
int main(void)
{
    int number1, number2, i ,j, flag;
    printf("Enter the two intervals:");
    scanf("%d     %d",&number1,&num
    ber2);
    printf("prime number between %d
    and %d:", number1,number2);
    for(i=number1+1;i<num
    ber2;i++)
    {
        flag=0;
        for(j=2;j<=i/2;++j)
        {
            if(i%j==0)
            {
                flag=1;
                break;
            }
        }
    }
    if(flag==0)
    printf("%d",i);
    return 0;
}
```

20. Write a C program to print the following pattern using loop:

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

A:
```
#include <stdio.h>
int main()
{
    int i, j, N;
    printf("Enter N: ");
    scanf("%d", &N);
    for(i=1; i<=N; i++)
    {
        for(j=1; j<=i; j++)
        {
            if(j % 2 == 1)
            {
                printf("1");
            }
            else
            {
                printf("0");
            }
        }
        printf("\n");
    }
    return 0;
}
```

21. Find the output:

```
{
    int a=1, b=1;
    while (a++ < 5)
    b*=a;
    printf("a=%d, b=%d",a,b);
}
```

A: Output: a=6, b=120

22. Find the values of x and y given by the following statements:

```
int x = 1, y=0;
y = x++ + ++x;
```

A: A: x =3, y=4