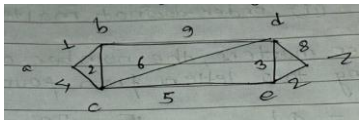
Group B

Core Programming

1. Write a program to calculate the sum of the first N natural numbers.
2. Write a program that prints numbers from 1 to N. Print 'Code' if the number is divisible by 3, 'Olympiad' if divisible by 5, 'CodeOlympiad' if divisible by both, and the number otherwise.
3. Write an algorithm to print every other element of an array starting from the first element.
4. Given an integer X, reverse its digits without converting it into a string.
5. Given a number, repeatedly sum its digits until the result is a single digit.
6. Write a program to replace all vowels in a given string with a specified character.
7. Write a program to check whether a given number N is a power of two. A number is a power of two if it can be written as 2^k for some integer k.
8. Write a program to count the number of positive, negative, and zero elements in an array.
9. From a given string, return the first non-repeating character. If none exists, return -1.
10. Write an algorithm to find the largest digit in a given number.

Writing

1. Write an Algorithm for Push operation in Stack.
2. Solve the recurrence relation : an= an-1 + 2 subject to initial condition a1 = 3.
3. Write down the expansion of (1+y)6 using Pascal's Triangle theorem.
4. Write down the steps for Krushkal's Algorithm.
5. Construct Binary Search Tree (BST) from following data : 25, 19, 21, 12, 18, 11, 9, 30.
6. Consider the set {a, b, c, d}.In how many ways can we select two of these letters (repetition is not allowed)
 i) Order Matters ii) Order Doesn't Matter
7. Find the GCD of 20 and 28 using Euclidean algorithm.
8. Use Dijkstra's Algorithm to find the length of the shortest path between the vertices a and z in the weighted graph displayed below .



Output Prediction

```c
#include <stdio.h>
int main()
{
    int a = 5, b = 3;
    printf("%d\n", a & b);
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int x = 1;
    printf("%d %d\n", x++, ++x);
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int num = 2;
    switch (num)
    {
    case 1:
        printf("One ");
    case 2:
        printf("Two ");
    case 3:
        printf("Three ");
    default:
        printf("Default");
    }
    return 0;
}
```

```c
#include <stdio.h>
void fun()
{
    static int count = 0;
    printf("%d ", ++count);
}
int main()
{
    fun();
    fun();
    fun();
    return 0;
}
```

```c
#include <stdio.h>
void fun(int a, int b)
{
    a += b;
    b += a;
    printf("%d %d\n", a, b);
}
int main()
{
    int x = 5, y = 10;
    fun(x, y);
    printf("%d %d\n", x, y);
    return 0;
}
```

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

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char str1[] = "Hello";
    char str2[] = "Hello";
    if (str1 == str2)
        printf("Equal\n");
    else
        printf("Not Equal\n");
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    for (int i = 0; i < 2; i++)
    {
        int i = 10;
        printf("%d ", i);
        i++;
    }
    return 0;
}
```

```c
#include <stdio.h>
void fun()
{
    static int x = 0;
    if (x < 5)
    {
        printf("%d ", x);
        x++;
        fun();
    }
}
int main()
{
    fun();
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int i = 0;
    while (i = 0)
    {
        printf("Looping\n");
    }
    return 0;
}
```

Design