

a) 1) To add first 10 terms of series using for
 $1/1! + 2/2! + 3/3! + \dots$

program:-

```
#include <bits/stdc++.h>
using namespace std;
double sum(double sum1)
{
    double result=0, factorial=1;
    for(int i=1; i<=sum1; i++)
    {
        factorial = factorial * i;
        result = result + (i/factorial);
    }
    return result;
}
int main()
{
    double n;
    cin >> "Enter the value of n" >> n;
    cout << "Sum: " << sum(n);
    return 0;
}
```

b) To print fibonacci series:-

function to print those numbers (pseudocode):-

```
void printseries(int n)
{
    int r=0, r1=1, i;
    if (n<1)
        return;
    cout << r << " ";
    for (i=1; i<=n; i++) {
        cout << r1 << " ";
        int next = r+r1;
        r = r1;
        r1 = next;
    }
}
```

c) Matrix pseudocode

```
void matrixoperations (int row, int column, int n)
```

```
{    int A[10][10];
```

```
    cout << "Enter the no. of rows & columns & integer n" << endl;
```

```
    cin >> row >> column >> n;
```

```
    for (int i=0; i<row; i++)
```

```
        for (int j=0; j<column; j++)
```

```
            cin >> A[i][j];
```

```
    cout << "you entered" << n;
```

```
    for (i=0; i<row; i++)
```

```
        { for (j=0; j<column; j++)
```

```
            cout << A[i][j] << " ";
```

```
            cout << "\n";
```

```
        }
```

```
    for (i=0; i<row; i++)
```

```
        { for (j=0; j<column; j++)
```

```
            {
```

```
                A[i][j] = n + A[i][j];
```

```
            }
```

```
        }
```

```
    }
```

d) To print the words of digits:-

```
void printvalue (char digit)
```

```
{
```

```
    switch (digit) {
```

```
        case '0': cout << "zero"; break;
```

```
        case '1': cout << "one"; break;
```

```
        case '2': cout << "two"; break;
```

```
        case '3': cout << "three"; break;
```

```
        case '4': cout << "four"; break;
```

```
        case '5': cout << "five"; break;
```

```
        case '6': cout << "six"; break;
```

```
        case '7': cout << "seven"; break;
```

```
        case '8': cout << "eight"; break;
```

```
        case '9': cout << "nine"; break;
```

```
    }
```

```
}
```

```
void wordprint (string n1) {
```

```
    int i, length = n1.length();
```

```
    for (i=0; i < length; i++) {
```

```
        printvalue(n1[i]);
```

```
    }
```

```
}
```

e) kmp algorithm to check / python implementation:-

```
def iswordpresent(sentence, word):
```

```
    word = word.upper()
```

```
    sentence = sentence.upper()
```

```
    lis = sentence.split()
```

```
    if (lis.count(word) > 0):
```

```
        return True
```

```
    else:
```

```
        return False
```

=> using kmp string matching algorithm also is beneficial.

2) Bug tracking system

