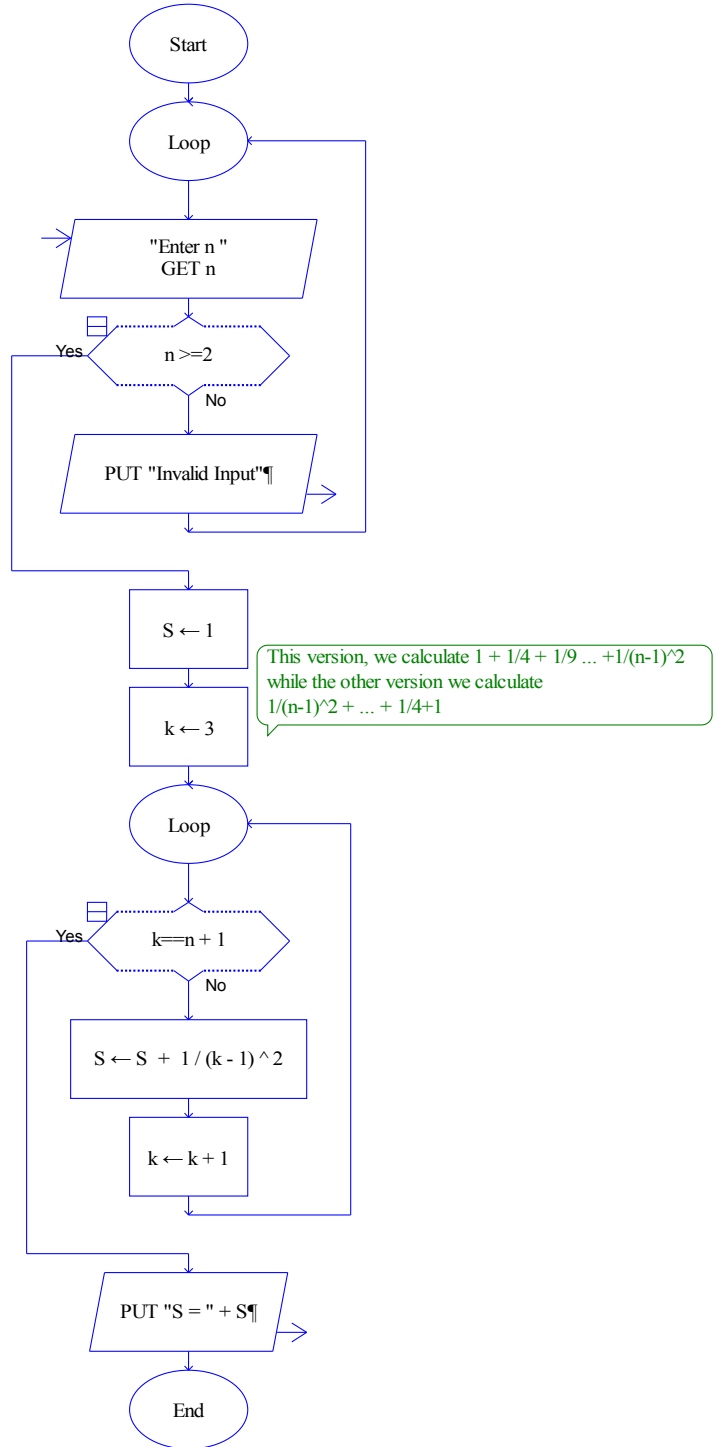
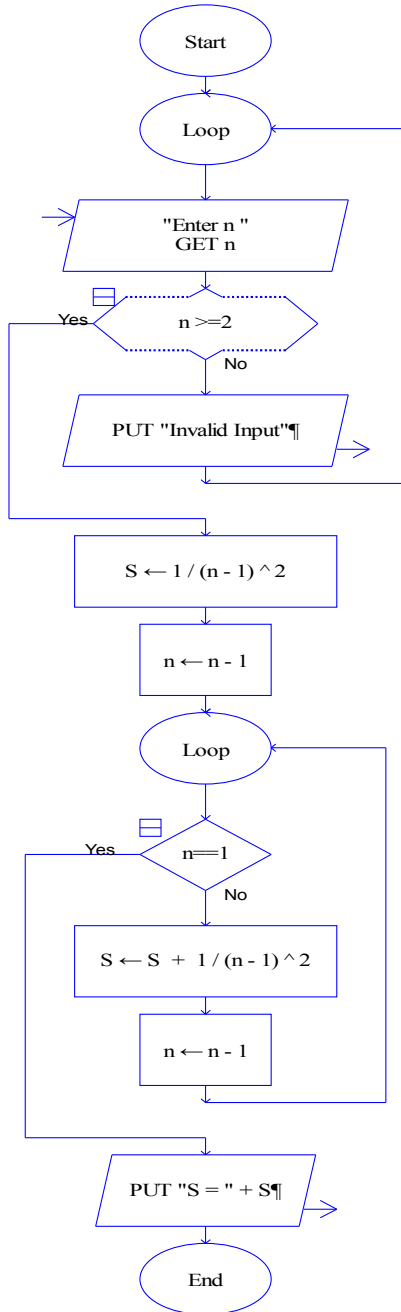


ECOR 1606 A - Midterm (Closed Books) - Duration 1h20

Problem 1 [4pt] : Write a flowchart which

(1) keeps requesting an integer $n \geq 2$ from the user (when the user enter a number < 2 , have the flowchart display "Invalid input" and request another value). Assume that the user will always enter an integer (and NOT a decimal) for n.

(2) calculates and displays the following sum : $S = \frac{1}{(2-1)^2} + \frac{1}{(3-1)^2} + \dots + \frac{1}{(n-1)^2}$



The two flowcharts are correct.

Problem 2 [3pt]: Convert the flowchart you had for Problem 1, into C++ program,

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    /* Beginning of your code */
    int n;
    double S; // S must be decimal and we must avoid divide two integers

    do{
        cout << "Enter n ";
        cin >> n;

        if( !(n>=2)){
            cout << "Invalid input" << endl;
        }
    }while( ! (n>=2) );

    S = 1;
    k=3;

    while( ! (k == n+1) ){
        S = S + 1.0 / ( (k-1) *(k-1) );
        // either the numerator or the denominator must be made a decimal
        // otherwise we get 0
        // the brackets in red are necessary
        k = k+1;
    }

    /*
    Instead of a while loop we could have a for loop

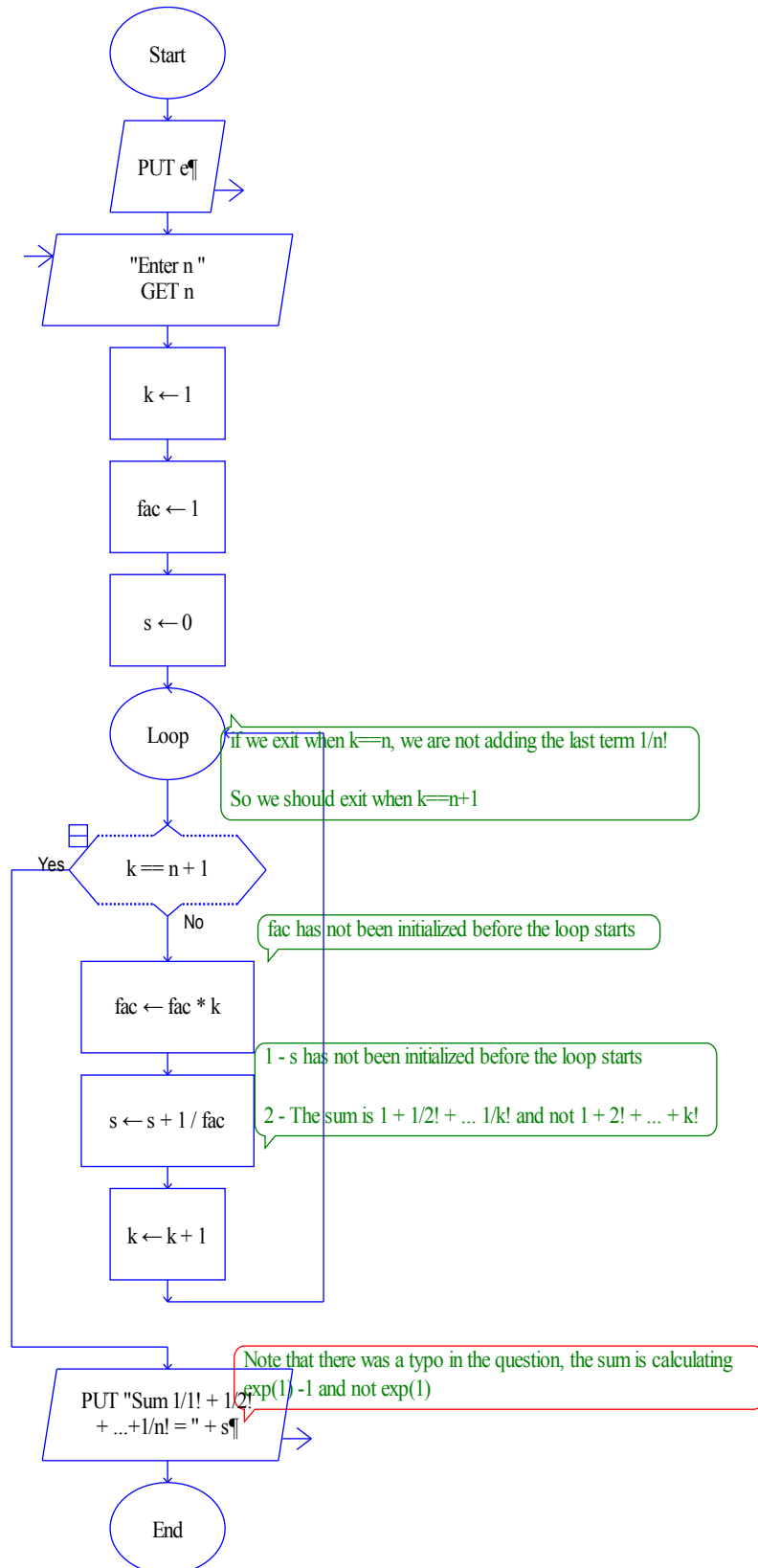
    for( k=3 ; !(k==n+1) ; k=k+1 ){
        S = S + 1.0 / ( (k-1) *(k-1) );
    }

    */
    cout << "S = " << S << endl;
    /* End of your code*/
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Problem 4 [3pt] : In this problem, we want to write a program which calculates :

$$s = \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!} , \text{ where } n \text{ is some value entered by the user. We will assume that the}$$

user always enters an integer $n > 0$. From the Lab 4, you may have recognized that when n is large enough, this sum is equal to $\exp(1) \sim 2.7183-1$. A student produced the following flowchart :



Marking scheme

| | |
|---|---|
| Problem 1 : | |
| Implementing the input to get n | 0.5 |
| Loop back when n is not valid | 0.5 |
| Displaying the "invalid input " message | 0.5 |
| Initializing S before the sum | 0.5 |
| Updating S inside the sum Writing this $\frac{1}{n}$ is not valid. Must write " 1/n ". | 0.75 |
| Updating the index (n or k) by incrementing or decrementing it inside the loop | 0.25 |
| Exit the loop with the correct condition | 0.5 |
| Displaying the final result | 0.5 |
| Problem 2 : | |
| Variable definition (S must be decimal, n could either int or float or double) | 0.5 |
| Input implementation | 0.25 |
| Display of invalid error message | 0.25 |
| Loop back when the input is invalid | 0.5 |
| Use of a loop either while or for loop to calculate the sum initialization of the index update of the index (increment or decrement) Writing n=k instead of n==k inside the loop condition is NOT VALID. Consider the loop incorrectly implemented | 0.5 |
| Update the sum s Note: the term $1 / ((k-1)*(k-1))$ MUST NOT be made such that it is a division of two integers. Note $(k-1)^2$ does not mean $(k-1)*(k-1)$. If the student uses <code>pow(k-1, 2)</code> without <code>#include <math.h></code> that is an error | 0.75 |
| display of final result <code>cout << "S = " + s</code> is NOT valid in C++ | 0.25 |
| Problem 3 | |
| Initialization of fac before loop is missing | 0.25 pt to point out the error 0.25 pt to correct it |
| Initialization of s before loop is missing | 0.25 pt to point out the error |

| | |
|--|---|
| | 0.25 pt to correct it |
| the condition to exit the loop is not $k==n$ but $k==n+1$ | 0.5 pt to point out the error 0.5 pt to correct it |
| update of s : it is not $s= s+ fac$ but $s=s+1/fac$ | 0.5 pt to point out the error 0.5 pt to correct it |