CST8233: Lab# 3

IEEE Floating Point Representation

Objectives

The objective of this lab is to get the student familiar with the theory components covered in week 2.

Earning

To earn your mark for this lab, each student should finish the lab's requirements within the lab session and demonstrate the working code to the instructor.

Laboratory Problem Description

Part A:

The following program demonstrate the overflow and underflow

Task A.1: Copy and paste the following code to your Visual Studio environment.

Task A.2: Run the code and notice the output of the program. Explain what happens to the instructor.

Task A.3: Change the loop's determinant "i" to few numbers higher than 127 and notice the output. Explain the results to the lab instructor.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

#include<math.h>
int main()
{
	int i;
	float n, x;
	n = 1.0;
	for (i = 0; i <= 127; i++)
	{
		n = n * 2.0;
		x = 1.0 / n;
			printf("%d %e %E\n", i, x, n);
	}

return 0;
}
```

Part B:

Truncation error occurs when the numerical methods used for solving mathematical problem. The numerical method approximates your function using a finite number of terms.
The difference between the exact and approximate solution is called truncation error. The following task demonstrate the truncation error to compute the famous factorial function. there is a famous numerical approximate formula, named after the Scottish mathematician James Stirling (1692-1770), that gives an approximate evaluation for n!.

$$\boldsymbol{n! = \sqrt{2 * \pi * n} * \ n^n * e^{-n}}$$

Task B :

Write a program to output a table with the following formate, you may use the factorial function from Lab#1 to calculate the analytical value of factorial function at each step.

n	***n*!**	**Stirling's**	**Absolute error**	**Relative error**

for $n = 1, 2, \ldots, 10$.

Hint: If your computer system does not have a predefined value for π,

Use either $\pi =$ `acos(-1.0)` or $\pi =$ `4.0*atan(1.0)`

Sample output:
Enter the number to calculate its factorial: 10

N	***n*!**	**Stirling's**	**Absolute error**	**Relative error**
1	**1**	**0.922137**	**0.077863**	**0.077863**
2	**2**	**1.919004**	**0.080996**	**0.040498**
3	**6**	**5.836210**	**0.163790**	**0.027298**
4	**24**	**23.506175**	**0.493825**	**0.020576**
5	**120**	**118.019168**	**1.980832**	**0.016507**
6	**720**	**710.078185**	**9.921815**	**0.013780**
7	**5040**	**4980.395832**	**59.604168**	**0.011826**
8	**40320**	**39902.395453**	**417.604547**	**0.010357**
9	**362880**	**359536.872842**	**3343.127158**	**0.009213**
10	**3628800**	**3598695.618741**	**30104.381259**	**0.008296**

Your numbers may be slightly different depending on the computer system and the precision used. Judging from the results, does the accuracy increase or decrease with increasing *n*.