CS540: Foundations of Computing System Design
Lab II Report

Submitted by-

Name: Prem Swarup
Roll No: 2003318
Branch: Mathematics & Computing (2024 batch)

Below are the results after calculating sum in c++ using formula:

$Sum = \sum_{i=1}^{\infty} \frac{1}{i} = 1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\ldots.$

S.No	'Sum' data type	'i' data type	Last 'i'	Last 'Sum'
1.	Int	Int	2	1
2.	Int	Float	2	1
3.	Float	Int	2	1.000000
4.	Float	Float	2097153	15.403683
5.	Double	Double	$937*10^{10}$	30.445704
6.	Float	Double	2097153	15.403683
7.	Double	Float	16777216	17.212748

Table 1: Data types of variables and corresponding value of sum and i till program stopped.

S.No.	Data type	Machine Epsilon
1.	int	0
2.	float	$1.192093 *10^{-07}$
3.	double	$1.110223 *10^{-16}$

Table 2: Machine epsilons of various data types used

C++ code to compute this:

```
#include<bits/stdc++.h>
using namespace std;

/* here type1 & type2 can be changed for choosing different data types for iterator and sum */
#define type1 float
#define type2 double

const long long divisor = 1e8 ;

template <typename T>
void check_epsilon(T one) {
	cout<<scientific ;
	T epsilon = 1 ;
	while(one + epsilon > one) {
		epsilon /= 2 ;
	}
	cout<<"machine epsilon: "<<epsilon<<endl ;
}


int main() {
	std::cout << std::fixed << std::setprecision(6);
	type1 i=1 ;
	type2 sum=0 ;
	type2 last=-1 ;

	while(sum!=last){
		last = sum ;
		sum += 1/i ;
		i++ ;
		if((long long)i%divisor==0){
			cout<<"iteration number: "<<i<<"   "<<"sum: "<<sum<<endl ;
		}
	}

	cout<<"final iteration number: <<i<<"   "<<"final sum: “<<sum<<endl ;
	return 0 ;
}
```

As we know that the “Sum” obtained from the above formula is divergent series that is the sum will never be bounded by any number and it will continue to grow as we keep adding the further terms of series. But due to precision limit of our computers, if we try to do it using our computers, it will eventually converge.

Case I.
In this case, data type of variable both ‘Sum’ & ‘i’ is int. So, when ‘i’ will become 2, 1/i would become 0. Hence sum will stop changing after ‘i’ becomes 2.

Case II.
In this case, data type of variable ‘Sum’ is int & ‘i’ is float. So, when ‘i’ will become 2, 1/i will become 0.5, but then it will be typecasted into ‘int’ data type before addition to sum, where value after decimal will be truncated and hence only 0 will be added to sum. So sum will stop changing after ‘i’ becomes 2.

Case III.
In this case, data type of variable both ‘Sum’ is float & ‘i’ is int. Here, when ‘i’ will become 2, 1/i would become 0. Hence sum will stop changing after ‘i’ becomes 2. In this implicit type casting from int to float will happen while addition of 1/i to the sum.

Case IV.
In this case, data type of variable both ‘Sum’ & ‘i’ is float. Here due to precision limit of float data type, additions of 1/i after i=2097153 becomes zero. So in code the value of sum of converges after 2097153 iterations of summation and final value of sum was 15.403683 . This is because float data type has limited precision which is usually about 7 decimal digits.

Case V.
In this case, data type of variable both ‘Sum’ & ‘i’ is double. In C++, there is also precision limit for double data type (which is usually around 17 digits) but that is too large to reach limit in one day of running code. (So I had to stop code around 12 hours because my laptop got too much heated.) After manually terminating the code the final registered value of ‘i’ was little above than $937*10^{10}$ and value of ‘Sum’ at that point was 30.445704 .

Case VI.
In this case, data type of variable ‘Sum’ is float and data type of variable ‘i’ is double. In this while addition of terms ‘1/i’ to the ‘Sum’ , at first ‘1/i’ is being calculated in double then the its value is implicitly type casted to float value while storing it in the ‘Sum’ and there it loses some precision. But since value of ‘1/i’ is calculated in double so ‘1/i’ only becomes 0 after very large amount of time. However, since float data type can hold upto only 7 significant digits of decimal points so sum stopped changing after 2097153 iterations and last noted value of sum was 15.403683.

Case VII.
In this case, data type of variable ‘Sum’ is double and data type of variable ‘i’ is float. In this while addition of terms ‘1/i’ to the ‘Sum’ , at first ‘1/i’ is being calculated in float then the its value is implicitly type casted to double value while storing it in the ‘Sum’ and since ‘Sum’ has higher precision than float so no loss of precision occurs. But since the maximum limit of float in my pc is 16777216, that is after ‘i’ become 16777216, i stopped changing and even after addition of 1 using code: `i += 1;` ‘i’ remained the same. So, sum was increasing by constant term which wasn’t correct, so I had to stop the code. The value of sum noticed after last iteration (i.e, when ‘i’ reached 16777216) was 17.212748.