

Output :--

```
vedu@pop-os: ~/assignment_DSA/prob_03
vedu@pop-os:~/assignment_DSA/prob_03$ gcc prob_03.c
vedu@pop-os:~/assignment_DSA/prob_03$ ./a.out
Enter the value of n(for single):... 15000

-----for float IEEE single precision-----
Sum upto 15000th value in HP is: 10.192990
Sum upto 15000th value in HP of reverse order is: 10.192993
error : -0.000003

-----for IEEE double precision-----
Sum upto 15000th value in HP is: 10.192988
Sum upto 15000th value in HP of reverse order is: 10.192988
error : 0.000000
vedu@pop-os:~/assignment_DSA/prob_03$
```

Explanation:--

Due to limited memory of data types.

Let's take an example float data type can store the value of pi up to 6 digits(3.141592) after the decimal point as its size 32bit and double data type has 64bit can store 14 digits(3.14159265358979), so in float, after six digits the info is lost due to limited size of it.

Here above same happens that much smaller operand couldn't participate as their actual values; that's why the error or difference occurs.