Arithmetic Exercise 5

Carly Smith

11 November 2020

1 Source Code

```
//Code to approximate e
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main() {
  int k;
  float e;
  int n;
  for (k = 1; k < 101; ++k){
      n = pow(10,k);
      e = pow((1+(1/n)),n);
      printf("%d \n",k);
      printf("%d \n",n);
      printf("%f \n", e);
    }
  return 0;
}
```

2 Explanation

This code prints k, n, and e on each loop iteration. I programmed it to print all three to better keep track of what the value were at each point in time. As n increases, the variable e gets increasingly close to the value of e. At k = 10, the single precision integer n overflows to -2147483648, rather than 10000000000.

3 Alternatives

To find an approximation of exp(x), a computer could use a finite number of terms from the taylor series approximation of exp(x). This method would be efficient because it can quickly evaluate an approximation of either e by itself or exp(x)

$$f(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

An approximation of e itself could also be found on its own by finding the value of exp(1) using the above sum evaluated at $x = 1$.