Lec06-control-Ryan-Sponzilli

September 12, 2024

1 ASTR 310 lecture 06: control

1.0.1 1. Machine precision

Because the internal representation of floating-point numbers uses a finite number of bits, there is a smallest number that when added to 1 produces a result different from 1. This value is sometimes called "machine epsilon (ϵ)". Write Python code to do the following. [5 pts]

- Choose an initial guess for ϵ, say 1.
- Create a loop that checks whether $\epsilon + 1$ is different from 1, and terminate if it is not.
- For each loop iteration, divide ϵ by 2.
- Report the final value of ϵ.

[1]:
```
e = 1
while (e + 1 != 1):
    e /= 2
print(e)
```

```
1.1102230246251565e-16
```

1.0.2 2. Fibonacci

The Fibonacci sequence is a sequence in which each number is the sum of the preceding two numbers: 1, 1, 2, 3, 5, 8, 13, and so on. Write Python code to generate the sequence. [5 pts] * Get an integer n from the user. * Begin with 1, 1. Print these two values. * Generate and print the remaining n values in the sequence.

[8]:
```
n = int(input("n = "))
fib = [1,1]
print(fib[0])
print(fib[1])
for i in range(n-2):
    fib.append(fib[-1] + fib[-2])
    print(fib[-1])
```

```
1
1
2
3
5
```

```
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
```

1.0.3 3. Primes: Sieve of Eratosthenes

The sieve of Eratosthenes is an old algorithm for finding all prime numbers smaller than some given integer n. Write Python code to implement the sieve: * Get n from the user. * Create a list L containing all the integers from 0 to n inclusive. You can do this by casting the result of the range() function to a list. * Let p = 2, the first prime. * Loop while p^2 is less than or equal to n: * Beginning with p^2, mark each of the multiples of p up to n in the list L (ie. for the multiple mp, set L[mp] = 0). * Beginning with p+1, find the first integer q such that L[q] != 0. If you reach the end of the list without finding such a number, terminate the loop. * Set p = q. * Print all the entries p > 1 for which L[p] != 0.

[40]:
```
n = int(input("n = "))
L = list(range(n-1))
p = 2
while (p**2 <= n):
    for m in range(p**2,n,1):
        if L[m-1] % p == 0:
            L[m-1] = 0
    for q in range(p+1,n,1):
        if (L[q] != 0):
            p = q
            break

for i in L:
    if L[i] != 0:
        print(L[i])
```

```
1
2
3
5
7
```

11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97