Khoi Duong

Prof. Yang

CE450L

12/13/2022

FINAL

GitHub link: <https://github.com/MynameisKoi/CE450L/tree/main/Final>



Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Final/1.py>

def PID():

    Kp = 8

    Ki = 5

    Kd = 1.6

    error = [30,20,10,7,5,2,1,-3,3]

    ctrl\_signal\_u = 394

    for i in range(2,len(error)):

        delta\_u = Kp\*(error[i] - error[i-1]) + Ki\*error[i] + Kd\*(error[i]-2\*error[i-1]+error[i-2])

        # format delta\_u 2 decimal points

        print('delta\_u\_', i, '=', "{:.2f}".format(delta\_u))

        ctrl\_signal\_u += delta\_u

        print('u\_', i, '=', "{:.2f}".format(ctrl\_signal\_u))

PID()

Run program & result:

Graphical user interface, text

Description automatically generated



Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Final/2.cpp>

#include <iostream>

#include <string>

#include <vector>

using namespace std;

int main(){

    int x[5];

    // set x[0] = 4500

    x[0] = 4500;

    for(int i=0; i<5; i++)

    {

        x[i] = i+1;

    }

    // print x

    cout << x << endl;

    cout << &x[0] << endl;

    cout << \*x << endl;

    cout << x[1] << endl;

    cout << x[2] << endl;

    cout << &x[1] << endl;

}

Although x[0] = 4500, after executing the code, we have x = [1, 2, 3, 4, 5]

The running result for the code:

Text

Description automatically generated

1. The value of x here will be the address of the first element in the list. Since in C++, the way we address the list is different from Python (in Python, print(x) will return the whole list x while in C++, it automatically returns the first element of the list x). In this case, the address is 0xd8f3bff660. Thus, it returns the address of the first element.
2. Reference to the first element of list x. In this case, the reference will be the address of the first element of the list, which is 0xd8f3bff660.
3. This will return the pointer to x, which means the value of the first element in the list. And since the list x = [1,2,3,4,5], it returns 1 (the first element)
4. The result for x[1] will be the value of the second element in the list x. Since the second value of the list x is 2, it will return the value 2.
5. The result for &x[1] will be the address of the second element in the list x. This is a reference to x[1]. In this case, the address of the second element in the list x is 4 bits more than the address of the first element in the list. Therefore, it returns 0xd8f3bff664.

Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Final/3.py>

print("Enter a number less than 10000: ", *end*="")

num = int(input())

# Display all prime number pair with difference of 2 between 1 and num

for i in range(1, num):

    # check prime number

    if i > 1:

        for j in range(2, i):

            if (i % j) == 0:

                break

        else:

            # check prime number pair with difference of 2

            if (i+2) > 1:

                for k in range(2, i+2):

                    if ((i+2) % k) == 0:

                        break

                else:

                    print(i, i+2)

Run program & result:

Text

Description automatically generated