For the following T(n) find values of n_0 and c such that cn^3 is larger than T(n) for all n larger than n_0.

$T(n) = n^3 - 5n^2 + 20n - 10$

Let n = 1
$(1)^3 - 5(1)^2 + 20(1) - 10 < C(1)^3$
$1 - 5 + 20 - 10 < C$
$6 < C$

$n_0 < n$; $n_0 = 0, -1, -2$
C = 7, 8, 9

Write a program that compares the values of y1 and y2 in the following expressions for values of n up to 100 in increments of 10.

Does the result surprise you?

y1 = 100 * n + 10
y2 = 5 * n * n + 2

```
Public Class Compare_valOfN_by10{
        Public static void main(String [] args) {
                Int n;
                Int y1;
                Int y2;
                System.out.print("%10s%10s%10sn", "n", "y1", "y2");
                For (n = 0; n <= 100; n += 10){
                        System.out.print("%10s%10s%10sn", n, 100 * n + 10, 5 * n * n + 2)
                }
        }
}
```

Not sure if y2 = 5 * n * n + 2 #1
Or y2 = 5 * n * (n + 2) #2

n	Y1	$Y2_1$	$Y2_2$
0	10	2	0
10	1010	502	600
20	2010	2002	2200
30	3010	4502	4800
40	4010	8002	8400
50	5010	12502	13000
60	6010	18002	18600
70	7010	24502	25200
80	8010	32002	32800
90	9010	39607	41400
100	10010	50002	51000

Write T(n) and big-O expressions for the following loops (explain how the result came about):

3.1.

```
for (int i = 0; i < n; i++)
        for (int j = 0; j < i * i; j++)
                cout << j << endl;
```

- -

$T(n) = n * n^2 = n^3$

Big-O : n^3

- -

3.2.

```
for (int i = n; i >= 0; i -= 2)
        cout << i << endl;
```

- -

n	ltr
7	3+1=4
6	3+1=4
5	2+1=3
4	2+1=3

- -

T(n) = (n/2).floor + 1

Big-O : n

- -

3.3.

```
for (int i = 0; i < n; i++)
        for (int j = i; j > 0; j /= 2)
                cout << j << endl;
```

- -

$T(n) = n * (Log_2 n + 1)$

Big-O : n Log(n)

- -

Show what happens in the figure below when the following statement executes:

v2.pop_back();

Answer: the Pop_back() function removes the last element and reduces the vector size by 1.

The new vector should be {1,2,3,4} with size of 9