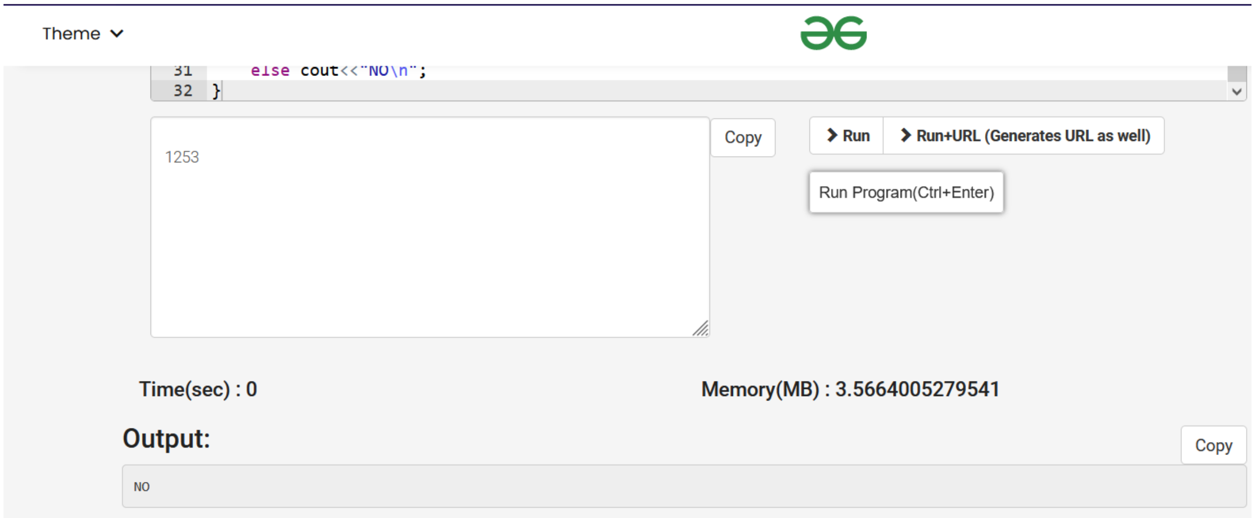
Assignment - 1

Q.1 Given a number x, determine whether the given number is Armstrong number or not. A positive integer of **n digits** is called an Armstrong number of **order n** (order is number of digits) if.
abcd... = pow(a,n) + pow(b,n) + pow(c,n) + pow(d,n) + ...

Ans:

```
#include<iostream>
#include<cmath>
#include<algorithm>
#define ll long long int
using namespace std;
int cnt(ll n){
        ll c=0;
        while(n>0){
                n=n/10;
                c++;
        }
    return c;
}
bool armstrong(ll n){
         ll x=cnt(n);;
         ll ans=n;
         ll sum=0;
        while(n>0){
                ll p=(n%10);
                sum+=(pow(p,x));
                n/=10;
        }
        if(sum==ans) return true;
        else return false;
}
main(){
        ll n; cin>>n;
           if(armstrong(n)) cout<<"YES\n";
        else cout<<"NO\n";
}
```



```
    else cout<<"NO\n";
}
```

```
1253
```

Time(sec) : 0

Memory(MB) : 3.5664005279541

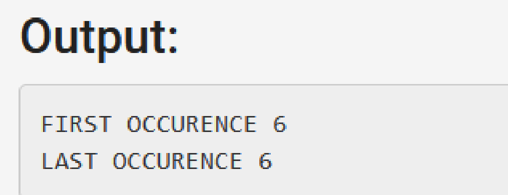
Output:

```
NO
```

Q.2 Given a sorted array with possibly duplicate elements, the task is to find indexes of first and last occurrences of an element x in the given array.

```
#include<iostream>

#include<vector>
```

```
#include<algorithm>

using namespace std;

void fun(vector<int> v,int x){

        int  a= lower_bound(v.begin(),v.end(),x)-v.begin();

        cout<<"FIRST OCCURENCE "<<a<<"\n";

        int  b= upper_bound(v.begin(),v.end(),x)-v.begin();

        cout<<"LAST OCCURENCE "<<b-1;

}

main(){

        vector<int> v={1, 3, 5, 5, 5, 5, 7, 123, 125};

        int x=7;

        fun(v,x);

}
```



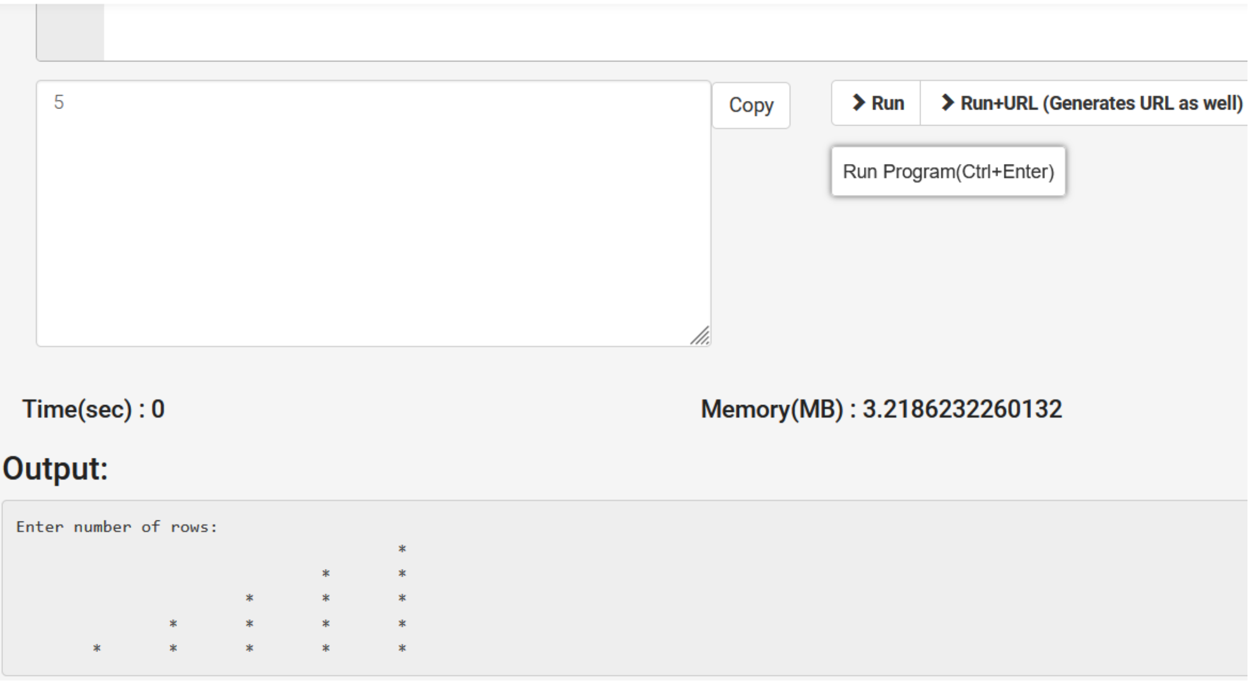
Output:

```
FIRST OCCURENCE 6
LAST OCCURENCE 6
```

Q.3 1. You are given a number n.
2. You've to create a pattern of * and separated by tab as shown in output format.

```
#include <iostream>

using namespace std;

int main()
```

```
{
    int n;
    cout << "Enter number of rows:"<<endl;
    cin >> n;
    for(int i = 1; i <= n; i++)
    {
        for(int j =n-1; j>= i; j--)
        {
            cout <<" "<<'\t';
        }
        for(int k=1;k<=i;k++)
        {
            cout<<'\t'<<"*";
        }
        cout <<endl;
    }
    return 0;
}
```



```
5
```

Time(sec) : 0

Memory(MB) : 3.2186232260132

Output:

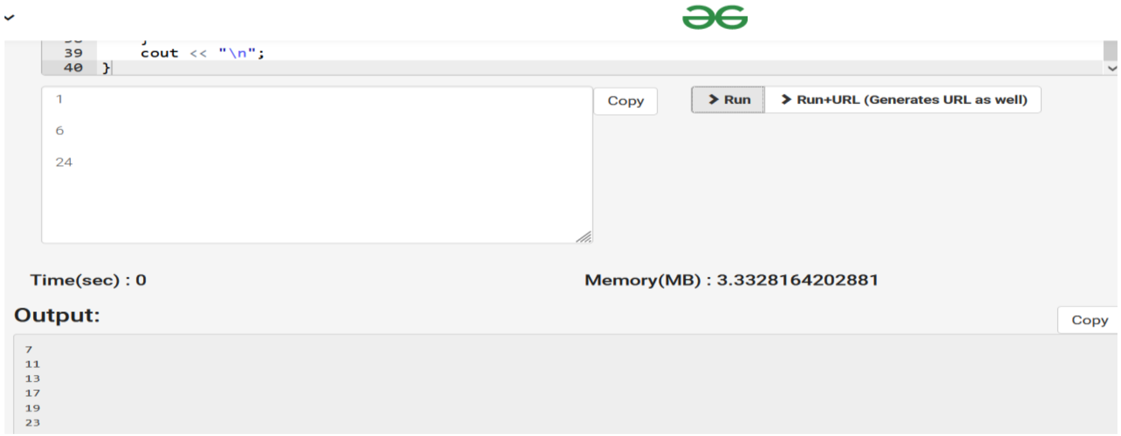
```
Enter number of rows:
                                        *
                                *       *
                        *       *       *
                *       *       *       *
        *       *       *       *       *
```

Q.4 1. You've to print all prime numbers between a range.
2. Take as input "low", the lower limit of range.
3. Take as input "high", the higher limit of range.
4. For the range print all the primes numbers between low and high (both included).

```
#include<iostream>

#include<algorithm>

using namespace std;

void solve();

int main() {

        int t;

        cin >> t;

        while (t--) {
```

```
        solve();
        }
        return 0;
}
void solve() {
        int m, n;
        cin >> m >> n;
        vector<bool> isPrime(n - m + 1, true);
        if (m == 1) isPrime[0] = false;
        for (int i = 2; i * i <= n; i++) {
        int firstMultiple = m / i * i;
        if (firstMultiple < m) {
                firstMultiple += i;
        }
        for (int j = max(firstMultiple, i * i); j <= n; j += i) {
                isPrime[j - m] = false;
        }
        }
        for (int i = m; i <= n; i++) {
        if (isPrime[i - m]) {
                cout << i << "\n";
        }
        }
```

```
	cout << "\n";
}
```



```
    cout << "\n";
}
```

```
1

6

24
```

Time(sec) : 0

Memory(MB) : 3.3328164202881

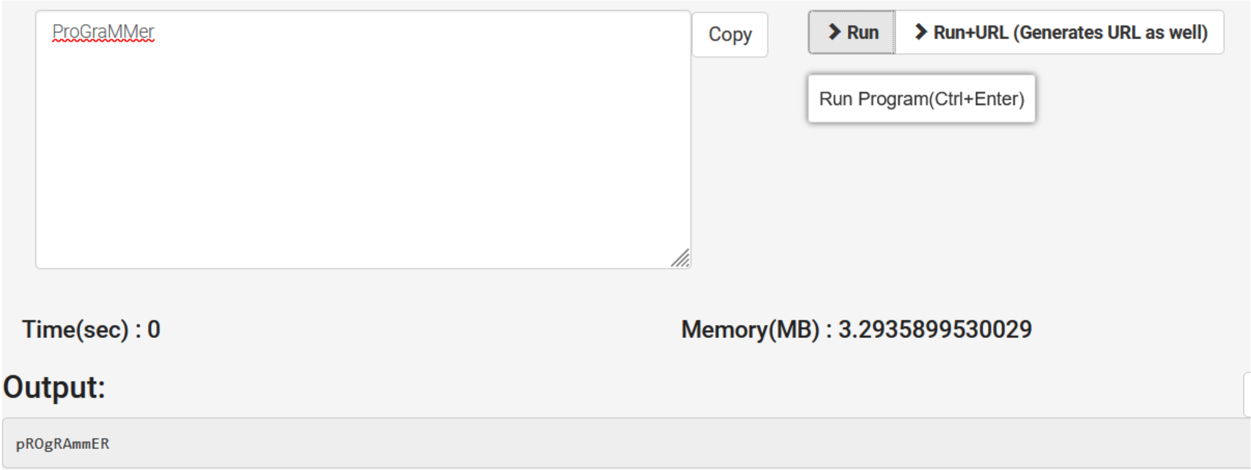
Output:

```
7
11
13
17
19
23
```

Q.5 1. You are given a string that contains only lowercase and uppercase alphabets.

2. You have to toggle the case of every character of the given string.

```
#include<iostream>

using namespace std;

main(){

	string s; cin>>s;

	for(int i=0;i<s.length();i++){

	if(s[i]>=65 && s[i]<=90) s[i]=s[i]+32;

	else s[i]=s[i]-32;

	}

	cout<<s;

}
```



ProGraMMer

Time(sec) : 0

Memory(MB) : 3.2935899530029

Output:

```
pROgRAmmER
```

NAME: Nisha Kumari