Problem: Given n, find the number of ones in the integers 1 – n.

```
# In Python

n=int(input("value? "))
s=0
for i in range(1,n+1):
   s+=str(i).count('1')
print(s)
```

```
//Using a C string

int main(){ char s [25];
int n;
  scanf("%d",&n);

 int s=0;
for (int i=1;i<=n;i++){
        itoa(i,s,10);
        s+=count(s,s+strlen(s),'1');
}
cout<<s<<endl;
}
```

```
//Using a C ++ string

#include<iostream>
#include<algorithm>
#include <stdlib.h>

using namespace std;

int main(){       char buffer [25];
        int n;
        cin>>n;
        int s=0;

        for (int i=1;i<=n;i++){
                string s(itoa(i,buffer,10));
                s+=count(s.begin(),s.end(),'1');
        }
        cout<<s<<endl;
}
```

```
#include<iostream>
using namespace std;

int main(){
	int n;
	cin>>n;
	int s=0;

	for (int i=1;i<=n;i++){
		int k=0;
		int j=i;
		while(j>0){
			if(j%10==1){
				k+=1;
			}
			j=j/10;
		}
		s+=k;
	}
cout<<s;

}
```

```
int main() {
    int n;
    cin>>n;

    int q = n, x = 1, s = 0;
     while (q > 0) {
        int digit = q % 10;
        q /= 10;
        s += q * x;
        if (digit == 1) s += n % x + 1;
        if (digit >  1) s += x;
        x *= 10;
    }
    cout<< s;
}
```