**RECURSION:**

[A - Print Many Hello World](https://vjudge.net/problem/Aizu-ITP1_3_A)#include <bits/stdc++.h>

using namespace std;

int c=0;

void printHello(){

if(c>=1000)

return;

cout<<"Hello World"<<endl;

c++;

printHello();

}

int main()

{

printHello();

return 0;

}

[B - The missing number](https://vjudge.net/problem/HackerRank-si-basic-the-missing-number)

 #include <bits/stdc++.h>

using namespace std;

int c=1;

int find(vector<int>& v){

if(v[c-1] != c){

return c;

}

c++;

return find(v);

}

int main()

{

int n=99;

vector<int> v(99);

for(int i=0;i<n;i++){

cin>>v[i];

}

sort(v.begin(),v.end());

cout<<find(v);

}

[C - Number reverse](https://vjudge.net/problem/HackerRank-si-basic-number-reverse)

#include <bits/stdc++.h>

using namespace std;

long int rev=0;

long int reverseNum(long int n){

if(n==0)

return rev;

int rem=n%10;

rev=rev\*10+rem;

return reverseNum(n/10);

}

int main()

{

long int n;

cin>>n;

long int res=reverseNum(abs(n));

if(n<0){

cout<<0-res<<endl;

}

else

cout<<res<<endl;

}

[D - Compute N!](https://vjudge.net/problem/HackerRank-si-basic-compute-n)

#include <bits/stdc++.h>

using namespace std;

long int factorial(long int n){

if(n==0)

return 1;

return n\*factorial(n-1);

}

int main()

{

int n;

cin>>n;

long int res=factorial(n);

cout<<res<<endl;

}

[E - Natural numbers sum](https://vjudge.net/problem/HackerRank-si-basic-natural-numbers-sum)

#include <bits/stdc++.h>

using namespace std;

long int sumOfEle(long int n){

if(n==1)

return 1;

return n+sumOfEle(n-1);

}

int main()

{

int n;

cin>>n;

long int res=sumOfEle(n);

cout<<res<<endl;

}

[F - Squares sum](https://vjudge.net/problem/HackerRank-si-basic-squares-sum)

#include <bits/stdc++.h>

using namespace std;

long int sumOfSqu(long int n){

if(n==1)

return 1;

return (n\*n)+sumOfSqu(n-1);

}

int main()

{

int n;

cin>>n;

long int res=sumOfSqu(n);

cout<<res<<endl;

}

[G - Cubes sum](https://vjudge.net/problem/HackerRank-si-basic-cubes-sum)

#include <bits/stdc++.h>

using namespace std;

long int sumofCubes(long int n){

if(n==1)

return 1;

return (n\*n\*n)+sumofCubes(n-1);

}

int main()

{

int n;

cin>>n;

long int res=sumofCubes(n);

cout<<res<<endl;

}

[H - Compute a power b.](https://vjudge.net/problem/HackerRank-si-basic-compute-a-power-b)

#include <bits/stdc++.h>

using namespace std;

long int power(int base,int expo){

if(expo==1)

return base;

return base\*power(base,expo-1);

}

int main()

{

int a,b;

cin>>a>>b;

if(b==0)

cout<<1<<endl;

else{

long int res=power(a,b);

cout<<res<<endl;

}

}

[I - Compute fibonacci number](https://vjudge.net/problem/HackerRank-si-basic-compute-fibonacci-number)

#include <bits/stdc++.h>

using namespace std;

int fibonocci(int n){

if(n==0 || n==1)

return n;

return fibonocci(n-1)+fibonocci(n-2);

}

int main()

{

int n;

cin>>n;

int res=fibonocci(n);

cout<<res<<endl;

}

[J - Check Armstrong number](https://vjudge.net/problem/HackerRank-si-basic-check-armstrong-number)

#include <bits/stdc++.h>

using namespace std;

long int temp=0;

long int armstrong(long int n,int t){

if(n==0)

return 0;

int rem=n%10;

temp+=pow(rem,t);

return armstrong(n/10,t);

}

int main()

{

long int n,res;

cin>>n;

int t=n;

int count=0;

while(t != 0){

count++;

t=t/10;

}

res=armstrong(n,count);

if(temp==n){

cout<<"Yes"<<endl;

}

else

cout<<"No"<<endl;

}

[K - Narcissistic numbers](https://vjudge.net/problem/HackerRank-si-basic-narcissistic-numbers)

#include <bits/stdc++.h>

using namespace std;

long int temp=0;

long int armstrong(long int n,int t){

if(n==0)

return 0;

int rem=n%10;

temp+=pow(rem,t);

return armstrong(n/10,t);

}

int main()

{

long int n,res;

cin>>n;

int t=n;

int count=0;

while(t != 0){

count++;

t=t/10;

}

res=armstrong(n,count);

if(temp==n){

cout<<"Yes"<<endl;

}

else

cout<<"No"<<endl;

}

[L - Print multiplication table](https://vjudge.net/problem/HackerRank-si-basic-print-multiplication-table)

 #include <bits/stdc++.h>

using namespace std;

long int temp=1;

void printTable(int n){

if(temp>10)

return;

cout<<n<<" \* "<<temp<<" = "<<n\*temp<<endl;

temp++;

printTable(n);

}

int main()

{

long int n;

cin>>n;

printTable(n);

}

[M - Reverse array](https://vjudge.net/problem/HackerRank-si-basic-reverse-array)

#include <bits/stdc++.h>

using namespace std;

long int temp=1;

void printArray(vector<long int>& v,int n){

if(n<0)

return;

cout<<v[n]<<" ";

printArray(v,n-1);

}

int main()

{

int n;

cin>>n;

vector<long int> v(n);

for(int i=0;i<n;i++){

cin>>v[i];

}

printArray(v,n-1);

}

[N - Sum of all odd elements](https://vjudge.net/problem/HackerRank-si-basic-sum-of-odd-elements)

#include <bits/stdc++.h>

using namespace std;

long int temp=0;

void sumOddArray(vector<long int>& v,int n){

if(n<0)

return;

if(v[n]%2 != 0)

temp+=(v[n]);

sumOddArray(v,n-1);

}

int main()

{

int n;

cin>>n;

vector<long int> v(n);

for(int i=0;i<n;i++){

cin>>v[i];

}

sumOddArray(v,n-1);

cout<<temp<<endl;

}

[O - Linear search on array](https://vjudge.net/problem/HackerRank-si-basic-linear-search-on-array)

#include <bits/stdc++.h>

using namespace std;

long int ind=-1;

long int searchArray(vector<long int>& v,int n,int key){

if(n<0)

return ind;

if(v[n]==key){

ind=n;

return ind;

}

return searchArray(v,n-1,key);

}

int main()

{

int n,key;

cin>>n>>key;

vector<long int> v(n);

for(int i=0;i<n;i++){

cin>>v[i];

}

cout<<searchArray(v,n-1,key)<<endl;

}