Problem Format

- Problem title
- Problem description
- Input specifications
- Output specifications
- Sample Input
- Sample Output

H. A + B Strikes Back

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A + B is often used as an example of the easiest problem possible to show some contest platform. However, some scientists have observed that sometimes this problem is not so easy to get accepted. Want to try?

Input

The input contains two integers a and b ($0 \le a, b \le 10^3$), separated by a single space.

Output

Output the sum of the given integers.

Sample test(s)

input
5 14
output
19

input
381 492
output
873

Online Judges

CODEFORCES	codeforces.com
[topcoder]™	topcoder.com/tc
Sphere online judge	spoj.com
UVa Online Judge	uva.onlinejudge.org
ACM-ICPC Live Archive	icpcarchive.ecs.baylor.edu
CODECHEF BETA	codechef.com
HackerRank	hackerrank.com
A2 Online Judge	a2oj.com

Input

C

```c
char c;          scanf("%c", &c);
char s[10];      scanf("%s", s);
int n;           scanf("%d", &n);
double v;        scanf("%lf", &v);
long long x;     scanf("%lld", &x);
char line[100]; gets(line);
```

C++

```cpp
char c;          cin >> c;
char str[10];    cin >> str;
string s;        cin >> s;
int n;           cin >> n;
double v;        cin >> v;
long long x;     cin >> x;
string line;     getline(cin, line);
```

Java

```java
Scanner sc   = new Scanner(System.in);
char c       = sc.next().charAt(0);
String s     = sc.next();
int n        = sc.nextInt();
double v     = sc.nextDouble();
long x       = sc.nextLong();
String line = sc.nextLine();
```

Input - Examples

```
5
12
0
98
-1
```

```
int n;
while (cin>>n && n!=-1){
    // work...
}
```

```
1 2 4 8 16
3 0 1
1
7 8 9 10
```

```
// #include <sstream>
string line;
int n, x, value[100];
while (getline(cin, line)){
    stringstream mycin(line);
    n = 0;
    while (mycin >> x){
        value[n] = x;
        ++n;
    }
    // work...
}
```

Input - Examples

```
Montaser Qasem 512
Mohammad Abu Dawwas 256
Ahmad 128
Abdullah Bahosain 1024
```

```
string line, name, tmp;
int number;
while (getline(cin, line)){
    stringstream mycin(line);
    name = "";
    while (mycin >> tmp){
        if(tmp[0]>='0' && tmp[0]<='9'){
            stringstream anothercin(tmp);
            anothercin >> number;
            break;
        }
        if (name.size() != 0)
            name += " ";
        name += tmp;
    }
    // work...
}
```

Input - Examples

```
Montaser Qasem 512
Mohammad Abu Dawwas 256
Ahmad 128
Abdullah Bahosain 1024
```

```
string line, name;
int number;
while (getline(cin, line)){
    for(int i=0;i<line.size();++i)
        if(line[i]>='0' && line[i]<='9'){
            name = line.substr(0, i - 1);
            stringstream abc(line.substr(i));
            abc >> number;
            break;
        }
    // work...
}
```

Input - Examples

```
Montaser Qasem 512
Mohammad Abu Dawwas 256
Ahmad 128
Abdullah Bahosain 1024
```

```c
char name[51];
int number;
while (scanf(" %[^0-9]", name) == 1){
    scanf("%d", &number);
    name[strlen(name) - 1] = 0;
    // work...
}
```

Output

```
     1...
    10...
   100...
  1000...
 10000...
100000...
```

```
for (int i = 1; i <= 100000; i *= 10)
    printf("%6d...\n", i);
```

```
1      ...
10     ...
100    ...
1000   ...
10000  ...
100000...
```

```
for (int i = 1; i <= 100000; i *= 10)
    printf("%-6d...\n", i);
```

```
3.48
3.478
```

```
double f = 3.478201;
printf("%.2lf\n", f);
printf("%.3lf\n", f);
```

Data Type Ranges

Data type	Size (byte)	Range
bool	1	True / False
char	1	-128 to 127
short	2	-32,768 to +32,767
int	4	-2,147,483,648 to +2,147,483,647
float	4	±3.4x10e-38 to ±3.4x10e38
double	8	±1.7x10e-308 to ±1.7x10e308
long long	8	-9.2e+18 to +9.2e+18

Double Vs. Float

```
                    float A = 3.141592653580;
                   double B = 3.141592653580;
printf("%.12f\n", A);  // 3.141592741013
printf("%.12lf\n", B); // 3.141592653580
                                    ^
```

Complexity

```
for (int i = 0; i<n; ++i){ // O( n )
    // ...
}

for (int i = 0; i<n; ++i){ // O( n*m )
    // ...
    for (int i = 0; i<m; ++i){
        // ...
    }
    // ...
}
```

Complexity

```
for (int i = 1; i<n; i *= 2){ // O( nlog(n) )
    // ...
    for (int j = 0; j<n; ++j){
        // ...
    }
    // ...
}

for (int i = 1; i<n; i *= 2){ // O( log(n)*log(m) )
    // ...
    for (int j = m; j>0; j /= 2){
        // ...
    }
    // ...
}
```

Complexity

```
char s[100001];
scanf("%s", s);
for (int i = 0; i<strlen(s); ++i){
    // ...
}

for (int i = 0; i<sqrt(n); ++i){
    // ...
}
```

Complexity

```
for (int i = 0; i*i <= n; ++i){
    // ...
}

for (int i = 0; i<n; ++i)
    for (int j = i; j<n; ++j){
        // ...
    }

for (int i = 0; i < n; i += 100){
    // ...
}

for (int i = 1; i <= n; ++i)
    for (int j = 1; j <= n; j += i){
        // ...
    }
```

STL - pair

```
pair<string, int> A;
cin >> A.first >> A.second;
A = make_pair("Amer", 80);

pair<string,pair<int,double> > A;
A.first = "Montaser";
A.second.first = 21;
A.second.second = 64.3;
```

pair Vs. struct

```
struct Student{
    string name;
    int age;
    double GPA;
};
Student A;
A.name = "Montaser";
A.age = 20;
A.GPA = 64.3;
```

References

- http://www.cpluspl
us.com/reference/