```cpp
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
    int a, b, c;
    cin >> a >> b >> c;
    cout << a + b + c;
    return 0;
}
```

```cpp
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
    int i; long l; char c; float f; double d;
    scanf("%d %ld %c %f %lf", &i, &l, &c, &f, &d);
    printf("%d\n%ld\n%c\n%.3f\n%.9lf", i, l, c, f, d);
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    if(n==1) cout<<"one";
    else if(n==2) cout<<"two";
    else if(n==3) cout<<"three";
    else if(n==4) cout<<"four";
    else if(n==5) cout<<"five";
    else if(n==6) cout<<"six";
    else if(n==7) cout<<"seven";
    else if(n==8) cout<<"eight";
    else if(n==9) cout<<"nine";
    else cout<<"Greater than 9";
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main() {
    int a, b;
    cin >> a >> b;
    for(int n=a;n<=b;n++){
        if(n==1) cout<<"one\n";
        else if(n==2) cout<<"two\n";
        else if(n==3) cout<<"three\n";
        else if(n==4) cout<<"four\n";
        else if(n==5) cout<<"five\n";
        else if(n==6) cout<<"six\n";
        else if(n==7) cout<<"seven\n";
        else if(n==8) cout<<"eight\n";
        else if(n==9) cout<<"nine\n";
        else if(n>9 && n%2==0) cout<<"even\n";
        else cout<<"odd\n";
    }
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int max_of_four(int a, int b, int c, int d){
    int m1=(a>b)?a:b;
    int m2=(c>d)?c:d;
    return (m1>m2)?m1:m2;
}
int main(){
    int a,b,c,d;
    cin>>a>>b>>c>>d;
    cout<<max_of_four(a,b,c,d);
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
void update(int *a,int *b){
    int sum=*a+*b;
    int diff=*a-*b;
    if(diff<0) diff=-diff;
    *a=sum;
    *b=diff;
}
int main(){
    int a,b;
    cin>>a>>b;
    update(&a,&b);
    cout<<a<<endl<<b;
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main(){
    int n;
    cin>>n;
    int arr[n];
    for(int i=0;i<n;i++){
        cin>>arr[i];
    }
    for(int i=n-1;i>=0;i--){
        cout<<arr[i]<<" ";
    }
    return 0;
}
```

```cpp
#include <sstream>
#include <vector>
#include <iostream>
using namespace std;
vector<int> parseInts(string str){
    vector<int> numbers;
    stringstream ss(str);
    int num;
    char ch;
    while(ss >> num){
        numbers.push_back(num);
        ss >> ch;
    }
    return numbers;
}
int main(){
    string str;
    cin>>str;
    vector<int> integers = parseInts(str);
    for(int i=0;i<integers.size();i++){
        cout<<integers[i]<<endl;
    }
    return 0;
}
```

```cpp
#include <iostream>
#include <unordered_set>
using namespace std;
int main(){
    long long N,S,P,Q;
    cin>>N>>S>>P>>Q;
    const long long MOD=(1LL<<31);
    unordered_set<long long> seen;
    long long a=S%MOD;
    seen.insert(a);
    for(int i=1;i<N;i++){
        a=(a*P+Q)%MOD;
        if(seen.find(a)!=seen.end()) break;
        seen.insert(a);
    }
```

```
    cout<<seen.size();
    return 0;
}

#include <exception>
#include <string>
#include <stdexcept>
#include <vector>
#include <cmath>
#include <iostream>
using namespace std;
class Server {
private:
    static int load;
public:
    static int compute(long long A,long long B){
        load+=1;
        if(A<0) throw invalid_argument("A is negative");
        vector<int> v(A,0);
        if(B==0) throw 0;
        int real=(A/B);
        int ans=v.at(B);
        return real+A-B*ans;
    }
    static int getLoad(){return load;}
};
int Server::load=0;
int main(){
    int T;cin>>T;
    while(T--){
        long long A,B;
        cin>>A>>B;
        try{
            cout<<Server::compute(A,B)<<endl;
        }catch(bad_alloc &){
            cout<<"Not enough memory"<<endl;
        }catch(invalid_argument &e){
            cout<<"Exception: "<<e.what()<<endl;
        }catch(exception &e){
            cout<<"Exception: "<<e.what()<<endl;
        }catch(...){
            cout<<"Other Exception"<<endl;
        }
    }
    cout<<Server::getLoad()<<endl;
    return 0;
}
```