

1 全体のテンプレ

別にこれに固定しなくてもよい

```
1  /*
2  input data
3  */
4
5  void init(){
6  }
7
8  bool input(){
9  }
10
11
12  /*
13  dp とか
14  */
15  void init_solve(){
16  }
17
18  void q_input(){
19  }
20
21  int solve(){
22      return 0;
23  }
24
25
26  int main(){
27      int m;
28      for(int i=0;i<m;i++){
29          cout<<solve()<<endl;
30      }
31  }
```

2 include

```
1  #include<cmath>
2  #include<cassert>
3  #include<string>
4  #include<vector>
5  #include<queue>
6  #include<stack>
7  #include<set>
8  #include<map>
9  #include<algorithm>
10 #include<utility>
11 using namespace std;
```

3 素数

```
1 vector<int> prime;
2 vector<int> prime_list;
3 void prime_set(int n){
4     n+=100;
5     prime.resize(n);
6     for(int i=0;i<n;i++){
7         prime[i]=1;
8     }
9     prime[0]=prime[1]=0;
10    for(int i=0;i<n;i++){
11        if(prime[i]){
12            for(int j=i*2;j<n;j+=i){
13                prime[j]=0;
14            }
15        }
16    }
17    for(int i=0;i<n;i++){
18        if(prime[i])prime_list.push_back(i);
19    }
20 }
```

4 文字列

```
1 string revStr(string s){
2     return string(s.rbegin(),s.rend());
3 }
```

5 二次元幾何

```
1 #define X first
2 #define Y second
3
4 typedef long long ll;
5 typedef pair<ll,ll> P;
6 int n;
7
8 int outer_product(P a,P b) {
9     return (a.X * b.Y - b.X * a.Y);
10 }
11
12 int inner_product(P a,P b){
13     return (a.X * b.X + a.Y * b.Y);
14 }
15
16 ll pow_len(P a) {
17     return a.X * a.X + a.Y * a.Y;
18 }
19
20 ll sqrt_len(P a){
21     return sqrt(pow_len(a));
22 }
23
24 P mkvec(P a,P b) {
25     return P(a.X - b.X, a.Y - b.Y);
26 }
```