

Enum party

one
two
three

Class candidate

public	
Candidate()	Default constructor,
Int get_ids()	Get the id of candidate
private	
Int id	Id fo candidate
String name	Name of candidate
Party party_name	Party of candidate

Class election

public:	
Election	Default constructor
Void clear()	Clear all when a win comes out
Void report()	Report each time after the campaign
Void report_win()	Report what party win
Int counting(district *)	Counting the constituents from given strict
register_candidate(candidate)	Register candidate before the campaign
go_campaigning()	Choose one candidate to do campaign
call_vote()	Candidates need to call for a vote
candidate* getCandidate()	Get candidate from outside
private:	

std::vetor<candidate> candidate_	The registered candidate
----------------------------------	--------------------------

### Class RepresentativeElection

public:	
representativeElection()	Default constructor
Int counting()	Count the vote in the district
private	
Party party_representative	The representative in the party

### Class district

public	
District()	Default constructor
Void change_party()	Change party of current district
Void get_constituent()	Get the number of constituent
Double get_squaremile()	Get square mile
private:	
Double square_mile	The number of a square mile
Party party_	The party belong of the district
std::vector<candidate> candidate_	All the candidates in the district
Party constitutent	Party of a constituent in the district

### Class electoralMap

public:	
electoralMap getInstanceMap()	Get the only map
District getDistrict(int)	Ge the district by id

friend std::ostream & operator<<(std::ostream& os, ElectoralMap map);	Output the map by <<
private	
ElectoralMap()	Initlize the map
static int count_district;	Set the index of district
std::map<int, district> map;	Map to store