

澳洲選舉 Australian Voting

4名候選人(ABCD)
11名投票人(第1,2,3,4志願)

例一:

選票	A	B	C	D
v01:	1	2	3	4
v02:	1	4	2	3
v03:	1	3	4	2
v04:	1	2	4	3
v05:	4	1	2	3
v06:	3	1	4	2
v07:	2	1	3	4
v08:	3	1	4	2
v09:	4	2	1	3
v10:	4	3	1	2
v11:	3	4	2	1

1 0

vote[0]	vote[1]	vote[2]	vote[3]
4	4	2	1

第1回合: 獲第1志願票總數

候選人 A B C D
獲選票 4 4 2 1

未有候選人得票超過半數，
淘汰得票最低的候選人D
並把選票過給第2志願候選人

vote[0]	vote[1]	vote[2]	vote[3]
4	4	3	0

第2回合: 獲第1志願票總數

候選人 A B C D
獲選票 4 4 3 X

第2回合:獲第1志願票總數

候選人 A B C D
獲選票 4 4 3 X

未有候選人得票超過半數，
淘汰得票最低的候選人C
並把選票過給第2,3志願候選人

選票	A	B	C	D		A	B	C	D
v09:	4	2	1	3	->	4	1	0	0
v10:	4	3	1	2	->	4	1	0	0
v11:	3	4	1	0	->	1	4	0	0

第3回合:獲第1志願票總數

候選人 A B C D
獲選票 5 6 X X

候選人B當選，得票超過半數

選票	A	B	C	D
v01:	1	2	3	4
v02:	1	4	2	3
v03:	1	3	4	2
v04:	1	2	4	3
v05:	4	1	2	3
v06:	3	1	4	2
v07:	2	1	3	4
v08:	3	1	4	2
v09:	4	1	0	0
v10:	4	1	0	0
v11:	1	4	0	0

例二: B當選

1	2	3	4
1	4	2	3
1	3	4	2
1	2	4	3
4	1	2	3
3	1	4	2
2	1	3	4
3	4	1	2
4	2	1	3
4	3	1	2
4	2	3	1

例三: Tie

1	4	3	2
1	2	3	4
3	1	2	4
4	1	2	3
4	2	1	3
2	3	4	1

n = 1324;

```
a = n/1000;  
b = n/100 %10;  
c = n/10 %10;  
e = n%10;
```

分拆
千位
百位
十位
個位

```
x      n      d  
a = 1324/1000;  
b =  324/100;  
c =   24/10;  
e =    4/1;
```

F4

1
3
2
4

```
n=1324;    A  
n=3124;    B  
n=3214;    C  
n=3241;    D
```

```
for(i=0;i<4;i++){  
    x =  
    printf("%i\n",x);  
    if(x== )  
        printf("%c", );  
    n =  
    d =  
}
```

```

int vnum[20] = { 1234, 1423, 1342, 1243, 4123,
                3142, 2134, 3142, 4213, 4312, 3421 };
int vote[4] = { 0 }, max = 11;
    
```

澳洲選舉 Australian Voting

```

void round1() { // 第一回合
    int i, n, d1000, d100, d10, d1;
    for (i=0; i<4; i++) vote[ ]=0;

    for (i=0; i<max; i++){
        printf("%i\n", vnum[i]);

        千位    d1000
        百位    d100
        十位    d10
        個位    d1

        if(d1000 == 1)
        else if(d100 == 1)
        else if(d10 == 1)
        else if(d1 == 1)
    }
}
    
```

```

void displayresults(){
    int i;
    for(i=0; ...
        printf...vote[ ]
    }
    
```

vote[0]	vote[1]	vote[2]	vote[3]
4	4	2	1

F4

```
int vnum[20]={ 1234,1423,1342,1243,4123,  
              3142,2134,3142,4213,4312,3421};  
int vote[4]={0}, max=11;
```

```
void round1() ...  
void displayresult() ...
```

未有候選人得票超過半數，
淘汰得票最低的候選人D
並把選票過給第2志願候選人

```
void knockout (int n){           // 0(A),1(B),2(C),3(D)  
    int i,d1000,d100,d10,d1;  
    for (i=0; i<max; i++){  
        }  
}
```

```
main() {  
    round1();           // 第1回合  
    displayresult();  
    knockout(3);       // 淘汰3(D)  
    round1();         // 第2回合  
    displayresult();  
}
```

n=0	1	2	3
1	2	3	4
1	4	2	3
1	3	4	2
1	2	4	3
4	1	2	3
3	1	4	2
2	1	3	4
3	1	4	2
4	2	1	3
4	3	1	2
3	4	2	1
1	0		

5

F4

```
void knockout (int n){ // 0(A),1(B),2(C),3(D)
    int i,d1000,d100,d10,d1;

    for (i=0; i<max; i++){
        printf("%i\n",vnum[i]);

        千位 d1000
        百位 d100
        十位 d10
        個位 d1

        if(n==3){
            if(d1==1){ ... }
        }else if(n==2){

        }else if(n==1){

        }else if(n==0){

        }

    }
}
```

n=0 1 2 3

1	2	3	4
1	4	2	3
1	3	4	2
1	2	4	3
4	1	2	3
3	1	4	2
2	1	3	4
3	1	4	2
4	2	1	3
4	3	1	2
3	4	2	1

1 0

6

```
int vote[4]={0}, max, out[4]={0};
char votestr[100][5]; // "1324",...
```

```
// File: AustralianVote.txt
```

```
int init() // load data into votestr[]
```

```
void round1()
```

vote[0]	vote[1]	vote[2]	vote[3]
4	4	2	1

```
void displayVotes()
```

```
void knockout(int n) // 0(A),1(B),2(C),3(D)
```

out[0]	out[1]	out[2]	out[3]
0	0	0	1

```
main(){
```

```
    max=init();
```

```
    round1();
```

```
    displayVotes();
```

```
    knockout(3); // (D)
```

```
    round1();
```

```
    displayVotes();
```

```
}
```

vote[0]	vote[1]	vote[2]	vote[3]
4	4	3	0

```
char votestr[100][5]; // "1324",...
int max=0;
```

F5

```
int init() {
    FILE *fp;
    char s[100];    int i=___;
    fp = fopen("AustralianVoting.txt","r");

    while(!feof(fp)){
        fgets(          );    s[strlen(s)-1]='\0';
        if(feof(fp)) break;

        i
    }
    fclose(fp);

    return ___;
}
```

```
main(){
    ___=init();
}
```

n=0 1 2 3

1	2	3	4
1	4	2	3
1	3	4	2
1	2	4	3
4	1	2	3
3	1	4	2
2	1	3	4
3	1	4	2
4	2	1	3
4	3	1	2
3	4	2	1

8

vote[0]	vote[1]	vote[2]	vote[3]
4	4	2	1

```

void round1(){
    int i,k;
    for (i=0;i<4;i++) vote[i]=0;

    for(k=0;k<max;k++) // electors
        for (i=0;i<4;i++) // choices(1234)
}

```

```

void displayresult(){
    int i;
    for (i=0;i<max;i++)
        printf("%s\n",votestr[i]);

    printf("\tA\tB\tC\tD\n\t");
    for (i=0;i<4;i++)
        printf("%i\t",          );
    printf("\n");
}

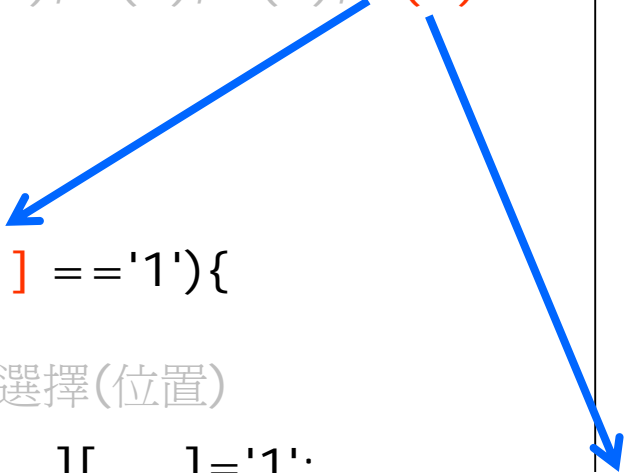
```

	i=0	1	2	3
k=0	1	2	3	4
	1	4	2	3
	1	3	4	2
	1	2	4	3
	4	1	2	3
	3	1	4	2
	2	1	3	4
	3	1	4	2
	4	2	1	3
	4	3	1	2
	3	4	2	1

```

void knockout (int n){ // n=0(A), 1(B), 2(C), 3(D)
    int i,k,x;
    out[ ]=1;
    for (k=0;k<max;k++){
        if(votestr[ ][ ] == '1'){
            x = 第2選擇(位置)
            votestr[ ][ ]='1';
            votestr[ ][ ]='0';
        }
    }
}

```



i=0 1 2 3

k=0

1	2	3	4
1	4	2	3
1	3	4	2
1	2	4	3
4	1	2	3
3	1	4	2
2	1	3	4
3	1	4	2
4	2	1	3
4	3	1	2
3	4	2	1

s="1324", c='2'

```

int findchr(char s[], char c) {
    int i;
    for (i=0; i<strlen(s); i++)
        if(____==c) return __;
    return -1; // not found
}

```