```cpp
//[集合堆疊電腦/The SetStack Computer]
#define IN "P0508IN.txt"
#define OUT "P0508OUT.txt"
//**************************************
#include <iostream>
#include <time.h>
using namespace std;
void redir(void);
//**************************************
/* Work Space*/
#include <string>
#include <set>
#include <map>
#include <vector>
#include <stack>
#include <algorithm> //set_union(), set_intersection()
#include <iterator> //inserter()

typedef set<int> Set;
map<Set, int> IdCache; //Set -> ID
vector<Set> SetCache; //ID -> Set

#define ALL(x) x.begin(),x.end()
#define INS(x) inserter(x,x.begin())

int ID(Set x);
//**************************************
int main(void)
{
    redir(); //redirection
//**************************************
/* Work Space*/
    int T, n;
    stack<int> s;//SetID


    scanf("%d", &T);
    while(T--){
        scanf("%d", &n);
        while(n--){
            string op; //block scope
            cin >> op;

            if(op[0] == 'P'){//PUSH
                Set x; //block scope
                s.push(ID(x));
            }else if(op[0] == 'D'){//DUP
                s.push(s.top());
            }else{
                Set x1, x2, x; //block scope
                x1 = SetCache[s.top()];
                s.pop();
                x2 = SetCache[s.top()];
                s.pop();

                if(op[0] == 'U'){//UNION
                    set_union(ALL(x1), ALL(x2), INS(x));
                }else if(op[0] == 'I'){//INTERSECT
                    set_intersection(ALL(x1), ALL(x2), INS(x));
                }else{//ADD
                    x = x2;
                    x.insert(ID(x1));
                }
```

```
64
65                  s.push(ID(x));
66              }
67              cout << SetCache[s.top()].size() << endl;
68          }
69          cout << "***" << endl;
70      }
71  //**************************************
72      freopen("CON", "r", stdin); //取消重新導向
73      freopen("CON", "w", stdout);
74
75      printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間
76
77      system("pause");
78      return 0; //the end...
79  }
80
81  void redir(void)
82  {
83      freopen(IN, "r", stdin);
84      freopen(OUT, "w", stdout);
85  }
86  //**************************************
87  /* Work Space*/
88  int ID(Set x)
89  {
90      if(IdCache.count(x)){
91          return IdCache[x];
92      }
93      SetCache.push_back(x);
94      return IdCache[x] = SetCache.size()-1;
95  }
96
97  //Input(IN) Sample
98  /*
99  2
100 9
101 PUSH
102 DUP
103 ADD
104 PUSH
105 ADD
106 DUP
107 ADD
108 DUP
109 UNION
110 5
111 PUSH
112 PUSH
113 ADD
114 PUSH
115 INTERSECT
116 */
```