//Assignment no 2

#include <iostream>

#define max 20

using namespace std;

class set

{

int A[max], cnt;

public:

set()

{

for(int i=0; i<max; i++)

{

A[i] = '\0';

}

cnt = 0;

}

void insert(void);

void display(void);

int search(int);

void size(void);

void del(void);

void union1(set, set);

set union2(set);

void intersection(set, set);

void minus(set, set);

void subset(set, set);

void reset(void);

};

void set :: insert(void)

{

int x;

cout << "\nEnter element to insert: ";

cin >> x;

A[cnt] = x;

cnt++;

}

void set :: display(void)

{

cout << "\nSet contents are as follows...";

for(int i=0; i<cnt; i++)

{

cout << A[i] << " ";

}

}

void set :: size(void)

{

cout << "\nThe size of the set is: " << cnt;

}

int set :: search(int x)

{

for(int i=0; i<max; i++)

{

if(A[i] == x)

{

cout << "\nElement found...";

return i;

}

}

return -1;

}

void set :: del(void)

{

int x, index;

cout << "\nEnter element to delete: ";

cin >> x;

index = search(x);

for(int i=index; i<max; i++)

{

A[i] = A[i+1];

}

cnt--;

}

void set :: union1(set m, set n)

{

for(int i=0; i<m.cnt; i++)

{

A[i] = m.A[i];

}

cnt = m.cnt;

for(int i=0; i<n.cnt; i++)

{

if(search(n.A[i]) == -1)

{

A[cnt] = n.A[i];

cnt++;

}

}

}

set set :: union2(set k)

{

set temp;

for(int i=0; i<cnt; i++)

{

temp.A[i] = A[i];

}

temp.cnt = cnt;

for(int i=0; i<k.cnt; i++)

{

if(search(k.A[i]) == -1)

{

temp.A[temp.cnt] = k.A[i];

temp.cnt++;

}

}

return temp;

}

void set :: intersection(set m, set n)

{

for(int i=0; i<n.cnt; i++)

{

if(m.search(n.A[i]) != -1)

{

A[cnt] = n.A[i];

cnt++;

}

}

}

void set :: minus(set m, set n)

{

for(int i=0; i<n.cnt; i++)

{

if(m.search(n.A[i]) == -1)

{

A[cnt] = n.A[i];

cnt++;

}

}

}

void set :: subset(set m, set n)

{

int scnt = 0;

for(int i=0; i<n.cnt; i++)

{

if(m.search(n.A[i]) != -1)

{

scnt++;

}

}

if(scnt == n.cnt)

{

cout << "\nB is a subset of A";

}

else

{

cout << "\nB is not a subset of A";

}

}

void set :: reset(void)

{

for(int i=0; i<cnt; i++)

{

A[i] = '\0';

}

cnt = 0;

}

int main()

{

set s, a, b;

int ch;

do

{

cout << "\n-------MENU-------";

cout << "\n1. Insert SET";

cout << "\n2. Insert A";

cout << "\n3. Insert B";

cout << "\n4. Display";

cout << "\n5. Size";

cout << "\n6. Search";

cout << "\n7. Delete";

cout << "\n8. Union 1";

cout << "\n9. Intersection";

cout << "\n10. Minus (A-B)";

cout << "\n11. Minus(B-A)";

cout << "\n12. Subset";

cout << "\n13. Reset";

cout << "\n14. Exit";

cout << "\nEnter your choice: ";

cin >> ch;

switch(ch)

{

case 1:

s.insert();

break;

case 2:

1. insert();

break;

case 3:

1. insert();

break;

case 4:

s.display();

1. display();
2. display();

break;

case 5:

s.size();

1. size();
2. size();

break;

case 6:

int x;

cout << "\nEnter element to search: ";

cin >> x;

s.search(x);

break;

case 7:

s.del();

break;

case 8:

s.union1(a, b);

break;

case 9:

s.intersection(a, b);

break;

case 10:

s.minus(a, b);

break;

case 11:

s.minus(b, a);

break;

case 12:

s.subset(a, b);

break;

case 13:

s.reset();

1. reset();
2. reset();

break;

case 14:

exit(0);

break;

default:

cout << "\nInvalid choice entered...";

}

}while(ch != 14);

return 0;

}

output:

gescoe@gescoe-OptiPlex-3010:~/Desktop/SE-A-55$ g++ set.cpp

gescoe@gescoe-OptiPlex-3010:~/Desktop/SE-A-55$ ./a.out

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 1

Enter element to insert: 1

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 1

Enter element to insert: 2

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 1

Enter element to insert: 3

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 2

Enter element to insert: 1

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 2

Enter element to insert: 2

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 1

Enter element to insert: 4

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 3

Enter element to insert: 6

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 3

Enter element to insert: 7

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 3

Enter element to insert: 8

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 5

The size of the set is: 4

The size of the set is: 2

The size of the set is: 3

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 7

Enter element to delete: 2

Element found...

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 8

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 4

Set contents are as follows...1 2 6 7 8

Set contents are as follows...1 2

Set contents are as follows...6 7 8

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 13

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 2

Enter element to insert: 1

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 2

Enter element to insert: 2

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 3

Enter element to insert: 2

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 9

Element found...

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 4

Set contents are as follows...2

Set contents are as follows...1 2

Set contents are as follows...2

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 10

Element found...

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 4

Set contents are as follows...2

Set contents are as follows...1 2

Set contents are as follows...2

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 12

Element found...

B is a subset of A

-------MENU-------

1. Insert SET
2. Insert A
3. Insert B
4. Display
5. Size
6. Search
7. Delete
8. Union 1
9. Intersection
10. Minus (A-B)
11. Minus(B-A)
12. Subset
13. Reset
14. Exit

Enter your choice: 14

[gescoe@gescoe-OptiPlex-3010](mailto:gescoe@gescoe-OptiPlex-3010):~/Desktop/SE-A-55$

It seems you want me to write or explain the code you've shared for **set operations**. Here's an overview of what your code does:

### Program Overview:

This program defines a **Set** class with operations that simulate various set operations like insert, delete, display, union, intersection, and more.

### Key Features:

* **Set Class:**
  + It includes an array A[max] to store elements and a counter cnt to keep track of the number of elements in the set.
  + It supports several operations like:
    - insert: Adds an element to the set.
    - display: Displays all elements of the set.
    - size: Displays the current size of the set.
    - search: Searches for an element in the set.
    - del: Deletes an element from the set.
    - union1: Combines two sets into one.
    - union2: Returns a new set that is the union of two sets.
    - intersection: Finds common elements between two sets.
    - minus: Returns elements that are in one set but not in another.
    - subset: Checks if one set is a subset of another.
    - reset: Clears the set.

### Functions and Their Explanations:

* **insert()**: Adds an element to the set.
* **display()**: Prints all elements in the set.
* **size()**: Displays the number of elements in the set.
* **search()**: Finds the index of an element in the set. If not found, returns -1.
* **del()**: Deletes an element from the set, adjusting the subsequent elements accordingly.
* **union1()**: Combines the current set with another set by inserting elements from the second set that aren't already in the current set.
* **union2()**: Creates and returns a new set that is the union of the current set and another set.
* **intersection()**: Finds and stores the common elements between two sets in the current set.
* **minus()**: Finds elements that are in the first set but not in the second set.
* **subset()**: Checks if all elements of the second set are present in the first set.
* **reset()**: Clears the current set by setting all its elements to \0 and resetting the counter.

### Example Run (Output):

The output provided shows the process of inserting, deleting, and performing set operations like union and size check. For example:

* The union of sets is performed successfully when the elements of one set are added to another if they do not already exist.
* The display function correctly lists the elements after operations like insertions and deletions.

If you need any further explanation or modifications, feel free to ask!