题目：集合交并差

班级：19184115

姓名：

学号：19063140

日期：2020.11.04

### 需求分析：

编制一个能演示执行集合的并、交和差运算的程序。集合的元素限定为小写字母字符[‘a’..’z’]，演示程序以用户和计算机的对话方式执行。

### 概要设计：

struct Set 集合用二进制表示集合中是否有字母字符

void SetInit(Set \*s) 集合初始化，为空集合

void SetInsert(Set \*s, char c) 在集合中插入一个字符

int SetFind(Set s, char c) 在集合中查询某个字符是否存在

void SetErase(Set \*s, char c) 删除集合中的某个字符

Set SetUnion(Set s1, Set s2) 集合求并

Set SetIntersection(Set s1, Set s2) 集合求交

Set SetDifference(Set s1, Set s2) 集合求差

void SetPrint(Set s) 集合输出

void SetRead(Set \*s) 读入一行字符集，构造一个集合

### 调试分析：

1. 遇到的问题
   1. 忘记初始化了,虽然本地测试可能看不出问题
   2. 发现答案错了，经过调试发现插入时写错了

改s->bit |= (c-'a') 为s->bit |= 1<<(c-'a')

1. 时空分析

时间复杂度为 O(1) 每种操作(除读入外)均为常数时间

空间复杂度为 O(1)

### 用户使用说明：

输入仅包含两行，每行若干个空格隔开的小写字母(可重复，自动去重

### 测试结果：

输入

a b c d e

c d e f g

输出

集合1与集合2的并为：{a,b,c,d,e,f,g}

集合1与集合2的交为：{c,d,e}

集合1与集合2的差为：{a,b}

集合2与集合1的差为：{f,g}

### 附录：

1. #include <stdio.h>
2. #include <stdlib.h>
3. #define N 100
5. **typedef** **struct** Set {
6. **int** bit;
7. } Set;
9. **void** SetInit(Set \*s) {
10. s->bit = 0;
11. }
13. **void** SetInsert(Set \*s, **char** c) {
14. s->bit |= 1<<(c-'a');
15. }
17. **int** SetFind(Set s, **char** c) {
18. **return** s.bit>>(c-'a')&1;
19. }
21. **void** SetErase(Set \*s, **char** c) {
22. s->bit &= ((1<<26)-1)-(1<<(c-'a'));
23. }
25. Set SetUnion(Set s1, Set s2) {
26. Set s;
27. s.bit = s1.bit|s2.bit;
28. **return** s;
29. }
31. Set SetIntersection(Set s1, Set s2) {
32. Set s;
33. s.bit = s1.bit&s2.bit;
34. **return** s;
35. }
37. Set SetDifference(Set s1, Set s2) {
38. Set s;
39. s.bit = s1.bit^(s1.bit&s2.bit);
40. **return** s;
41. }
43. **void** SetPrint(Set s) {
44. putchar('{');
45. **for** (**int** i = 0, first = 1; i < 26; ++i) {
46. **if** ((s.bit>>i)&1) {
47. **if** (!first) putchar(',');
48. putchar('a'+i);
49. first = 0;
50. }
51. }
52. putchar('}');
53. putchar('\n');
54. }
56. **void** SetRead(Set \*s) {
57. SetInit(s);
58. **char** c;
59. **while** ((c = getchar()) != '\n') {
60. **if** (c >= 'a' && c <= 'z') SetInsert(s, c);
61. }
62. }
64. **int** main() {
65. Set s1, s2;
66. printf("请在一行内输入集合1的各个元素：");
67. SetRead(&s1);
68. printf("请在一行内输入集合2的各个元素：");
69. SetRead(&s2);
70. printf("集合1与集合2的并为：");
71. SetPrint(SetUnion(s1, s2));
72. printf("集合1与集合2的交为：");
73. SetPrint(SetIntersection(s1, s2));
74. printf("集合1与集合2的差为：");
75. SetPrint(SetDifference(s1, s2));
76. printf("集合2与集合1的差为：");
77. SetPrint(SetDifference(s2, s1));
78. **return** 0;
79. }