Разработать ООП для объединения 2-х любых подмножеств латинских букв, которые передаются ей через аргументы командной строки. Полученный результат должен отображаться строкой стандартного вывода, где все буквы перечислены в алфавитном порядке. Программная реализация вычислений должна быть основана на разработке класса множества латинских букв. При этом состав множества кодируется двоичными разрядами целого числа без знака в приватном поле данных этого класса, а публичный компонентный метод предназначается для перегрузки оператора ‘**+’** в операции объединения 2-х множеств. Конструктор класса должен обеспечивать преобразование символьной записи множества в двоичный целочисленный формат. Необходимо также предусмотреть дружественную перегрузку оператора ‘**<<’** потока стандартного вывода или компонентный оператор преобразования типа множества в символьную строку для отображения объектов класса множества латинских букв.

1: #include <cctype>

2: #include <iostream>

3: using namespace std;

4:

5: class Alpha

6: {

7: private:

8: unsigned bin;

9: public:

10: Alpha() {bin=0;};

11: Alpha(char\*);

12: Alpha operator +(Alpha&);

13: operator char\*();

14: friend ostream& operator << (ostream&, Alpha&);

15: };

16:

17: Alpha::Alpha(char\* s)

18: {

19: bin=0;

20: while(\*s)

21: {

22: bin |= (1 << (tolower(\*s) - 'a'));

23: s++;

24: }

25: }

26:

27: Alpha Alpha::operator + (Alpha& y)

28: {

29: Alpha z;

30: z.bin = bin | y.bin;

31: return(z);

32: }

33:

34: ostream& operator << (ostream& out, Alpha& z)

35: {

36: unsigned bit=1;

37: int i;

38: for(i=0; i<26; i++)

39: {

40: if((z.bin & bit)>0)

41: out<<(char)('a'+i);

42: bit=bit<<1;

43: }

44: return out;

45: }

46:

47: Alpha::operator char\*()

48: {

49: static char s[32];

50: unsigned w=bin;

51: int i=0, j=0;

52: while(w > 0)

53: {

54: if(w & 1)

55: s[j++]='a'+i;

56: i++;

57: w = w >> 1;

58: }

59: s[j++]='\n';

60: s[j]='\0';

61: return (s);

62: }

63:

64: int main (int argc, char\* argv[])

65: {

66: Alpha x(argv[1]);

67: Alpha y(argv[2]);

68: Alpha z;

69: z=x+y;

70: cout<<z<<endl; // или cout<<(char\*)z;

71: return (0);

72: }