Скелет сета:

struct Node{  
 int data;  
 Node\*prev;  
 Node(int d): data(d), prev(nullptr){}  
};  
class Set\_Int {  
 Node\*top;  
public:  
 Set\_Int(Node\*t):top(t){}  
 void insert(int);  
 bool isInto(int);  
 int end();  
 Set\_Int operator++(int);  
 friend ostream &operator<<(ostream&, const Set\_Int&);  
 Node\* getTop();  
 Set\_Int():top(nullptr){}  
};  
int sum(Set\_Int set1, Set\_Int set2, Set\_Int res);

додаємо елемент в сет:

void Set\_Int::insert(int num) {  
 Node\*curr = top;  
 bool check = false;  
 int c = 0;  
 while(curr){  
 c++;  
 if(curr->data==num)  
 check = true;  
 curr = curr->prev;  
 }  
 if(!check){  
 int arr[c+1], ind = 0;  
 curr = top;  
 while(curr){  
 arr[ind++] = curr->data;  
 curr = curr->prev;  
 }  
 arr[ind] = num;  
 for(int i = 1;i<c+1;i++){  
 for(int j = i-1;j>=0;j--){  
 if(arr[j+1]<arr[j]){  
 int temp = arr[j+1];  
 arr[j+1] = arr[j];  
 arr[j] = temp;  
 }  
 }  
 }  
 top = nullptr;  
 for(int i = c;i>=0;i--){  
 Node\*newSet = new Node(arr[i]);  
 newSet->prev = top;  
 top = newSet;  
 }  
 }  
}

перевірка наявності елементу в сеті:

bool Set\_Int::isInto(int num) {  
 Node\*curr = top;  
 bool check = false;  
 while(curr){  
 if(curr->data==num)  
 check = true;  
 curr = curr->prev;  
 }  
 return check;  
}

перегруз++

Set\_Int Set\_Int::operator++(int) {  
  
 Node\*t = new Node(rand()%255+1);  
 this->insert(t->data);  
 return \*this;  
}

перегруз<<

ostream& operator<<(ostream& cout, const Set\_Int& n){  
 Node\*curr = n.top;  
 string res = "";  
 while(curr){  
 res+= to\_string(curr->data)+"\t";  
 curr = curr->prev;  
 }  
 cout<<res;  
 return cout;  
}

сума елементів що є в 2 але нема в 1

int sum(Set\_Int set1, Set\_Int set2, Set\_Int res){  
 Node\*curr = res.getTop();  
 int sum = 0;  
 while(curr){  
 if(set2.isInto(curr->data)&&!set1.isInto(curr->data))  
 sum+=curr->data;  
 curr = curr->prev;  
 }  
 return sum;  
}