# Assignment 4-Set Theory

Problem Statement: In Second year Computer Engineering class of M students, set A of students play cricket and set B of students play badminton. Write

C/C++ program to find and display-

1. Set of students who play either cricket or badminton or both
2. Set of students who play both cricket and badminton
3. Set of students who play only cricket
4. Set of students who play only badminton
5. Number of students who play neither cricket nor badminton

## Concepts Used

* Simple Array
* Set Theory

## Theory of Concepts Used

### Simple Array

An array in C++ Programing Language can be defined as number of memory locations, each of which can store the same data type and which can be references through the same variable name.

An array is a collective name given to a group of similar quantities. These similar quantities could be percentage marks of 100 students, number of chairs in home, or salaries of 300 employees or ages of 25 students. Thus an array is a collection of similar elements. These similar elements could be all integers or all floats or all characters etc. Usually, the array of characters is called a string, where as an array of integers or floats is called simply an array. All elements of any given array must be of the same type i.e we cant have an array of 10 numbers, of which 5 are ints and 5 are floats.

Arrays and pointers have a special relationship as arrays use pointers to reference memory locations.

Declaration of an Array

Arrays must be declared before they can be used in the program. Standard array declaration is as

type variablename[lengthofarray];

Here type specifies the variable type of the element which is going to be stored in the array. In C programmin language we can declare the array of any basic standard type which C language supports. For example

double height[10]; float width[20]; int min[9]; char name[20];

In C++ Language, arrays starts at position 0. The elements of the array occupy adjacent locations in memory. C Language treats the name of the array as if it were a pointer to the first element This is important in understanding how to do arithmetic with arrays. Any item in the array can be accessed through its index, and it can be accesed any where from with in the program. So

m=height[0]; variable m will have the value of first item of array height.

The program below will declare an array of five integers and print all the elements of the array.

int myArray [5] = {1,2,3,4,5}; To print all the elements of the array for (int i=0;i<5;i++){

cout<<myArray[i];

}

\subsubsection{Set Theory}

There are 4 basic set operations: union, intersection, complement, and difference. Perhaps the best way to understand them is to use what are called Venn diagrams.

1) Union. A B is the set that contains all the elements in either A or B or both:

A B = {x | x A or x B}.

For example, if A = { 1, 2, 3} and B ={ 3, 4, 5}, then A B = { 1, 2, 3,

4, 5}

2) Intersection. A B is the set that contains all the elements that are in both A and B:

A B = {x | x A and x B}.

For example, if A = { 1, 2, 3} and B ={ 3, 4, 5}, then A B = { 3 }

1. Complement. A’ is the set that contains everything in the universal set that is not in A:

A’ = {x | x U and x A}.

For example, if = { 2, 4, 6, 8, 10, 12} and A = { 2, 4}, then A’ ={ 6, 8, 10, 12}

1. Difference. A - B is the set that contains all the elements that are in A but not in B:

A - B = {x | x A and not x B}.

For example, if A = { 1, 2, 3} and B ={ 3, 4, 5}, then A - B = { 1, 2 }

## Algorithm

1. Intersection Algorithm.

Step 1 : Declare the array intersect[15] and the variables flag,k=0 ,i and j.

Step 2 : Use

Step 2 : Initialise the counter i=0 and repeat the the following steps till badm[i]!=\0

* 1. : Set flag to 0
  2. : Initialise the counter j=0 and repeat these steps till cric[j]!=\0

2.2.1 : If(cric[j] == badm[j]) Set flag =1

2.3 If(flag == 1)

Store the value of currently set badm array element into intersect[k] array. Increment k by 1.

1. Union Algorithm

Step 1 : Declare the variables i,j,k=0,flag=0, array unionArr[20] and use the already user entered arrays cric[] and badm[]

Step 2 : Copy the entire cric[] array into unionArr[]

Step 3 : Create a loop which repeats the following steps till badm[i]!=\0

* 1. : Set flag to 0
  2. : Create a loop which repeats the following steps till cric[j]!=\0 3.2.1 : If(cric[j] == badm[i]) Set flag to 1.

Increment j by 1

* 1. : If (flag!=1)

Store the value of currently set badm[] array into unionArr[k]

Increment k by 1

* 1. Increment i by 1

1. (AUB) ie Students who do not play/like any games

Step 1 : Create an Array M[100] which contains the roll numbers of every student.

Step 2 : Declare the variables flag=0, k=0 ,i=0, j=0and array neith[50].

Step 3 : Create a loop which repeats the following steps till M[i]!=\0

* 1. : Set flag to 0
  2. : Create a loop which repeats the following steps till unionArr[j]!=\0

3.2.1 : if(M[i] == unionArr[j]) Set flag to 1.

Increment the value of j by 1

* 1. : If(flag!=1)

Store the value of currently set M[i] into neith[k]

Increment the value of k by 1

* 1. Increment the value of i by 1
  2. Repeat if the loop condition is satisfied

1. A-B or (cric badm)

Step 1 : Declare the variables i=0,j=0,k=0,lyCric[20],flag=0

Step 2 : Create a loop which repeats the following steps till cric[i]!=\0

* 1. : Set flag to 0
  2. : Create a loop which repeats the following steps till badm[j]!=\0
     1. if(crick[i]==badm[j]

Set flag to 1

* + 1. : Increment the value of j by 1.
  1. : If(flag!=1)

Store the value of currently set cric[] element into lyCric[k]

Increment the value of k by 1

* 1. Increment the value of i by 1
  2. : Repeat the steps till the loop condition is satisfied

1. B-A or (badm-cric)

Step 1 : Declare the variables i=0,j=0,k=0,lyBadm[20],flag=0

Step 2 : Create a loop which repeats the following steps till badm[i]!=\0

* 1. : Set flag to 0
  2. : Create a loop which repeats the following steps till cric[j]!=\0
     1. if(crick[i]==badm[j]

Set flag to 1

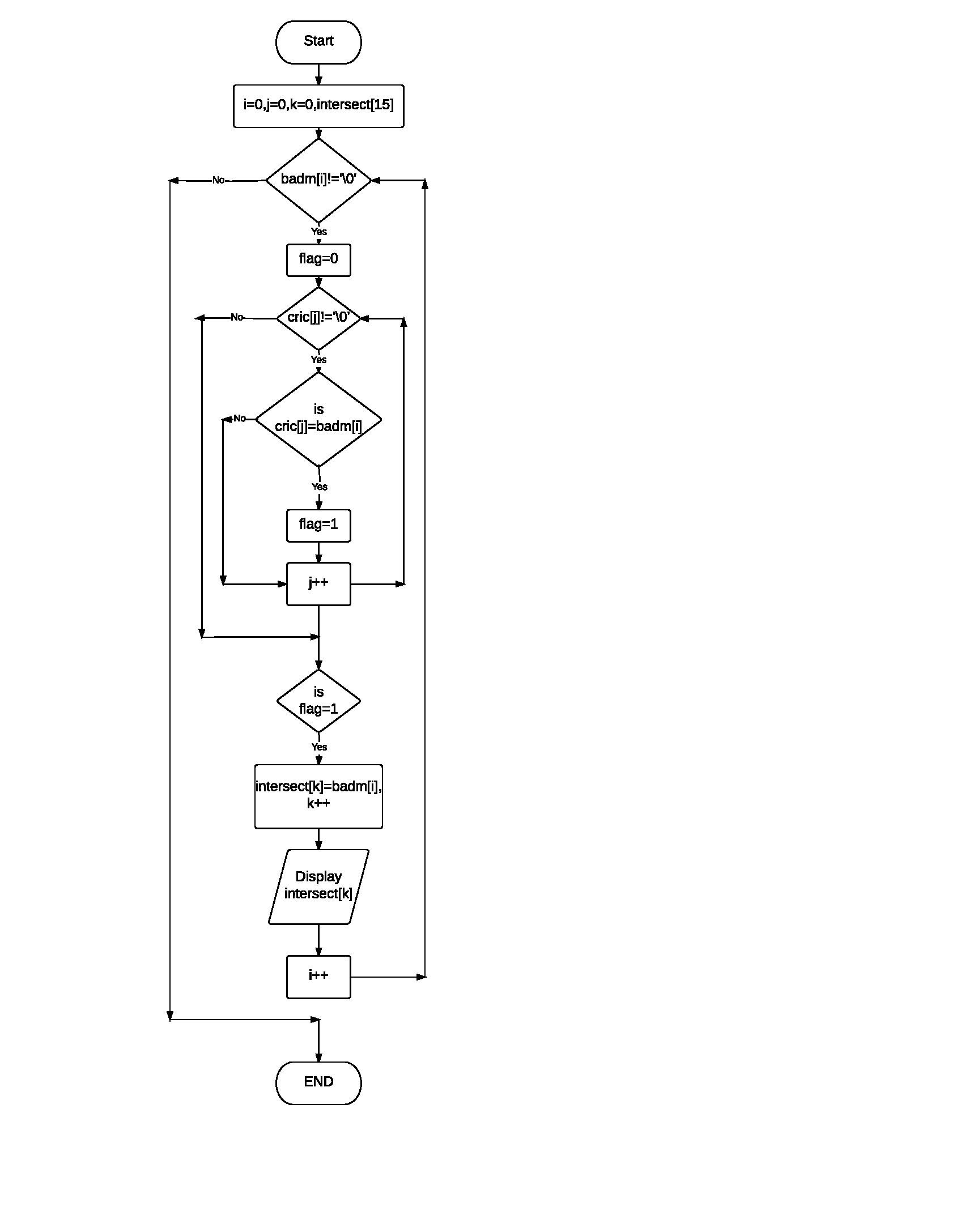
* + 1. : Increment the value of j by 1.
  1. : If(flag!=1)

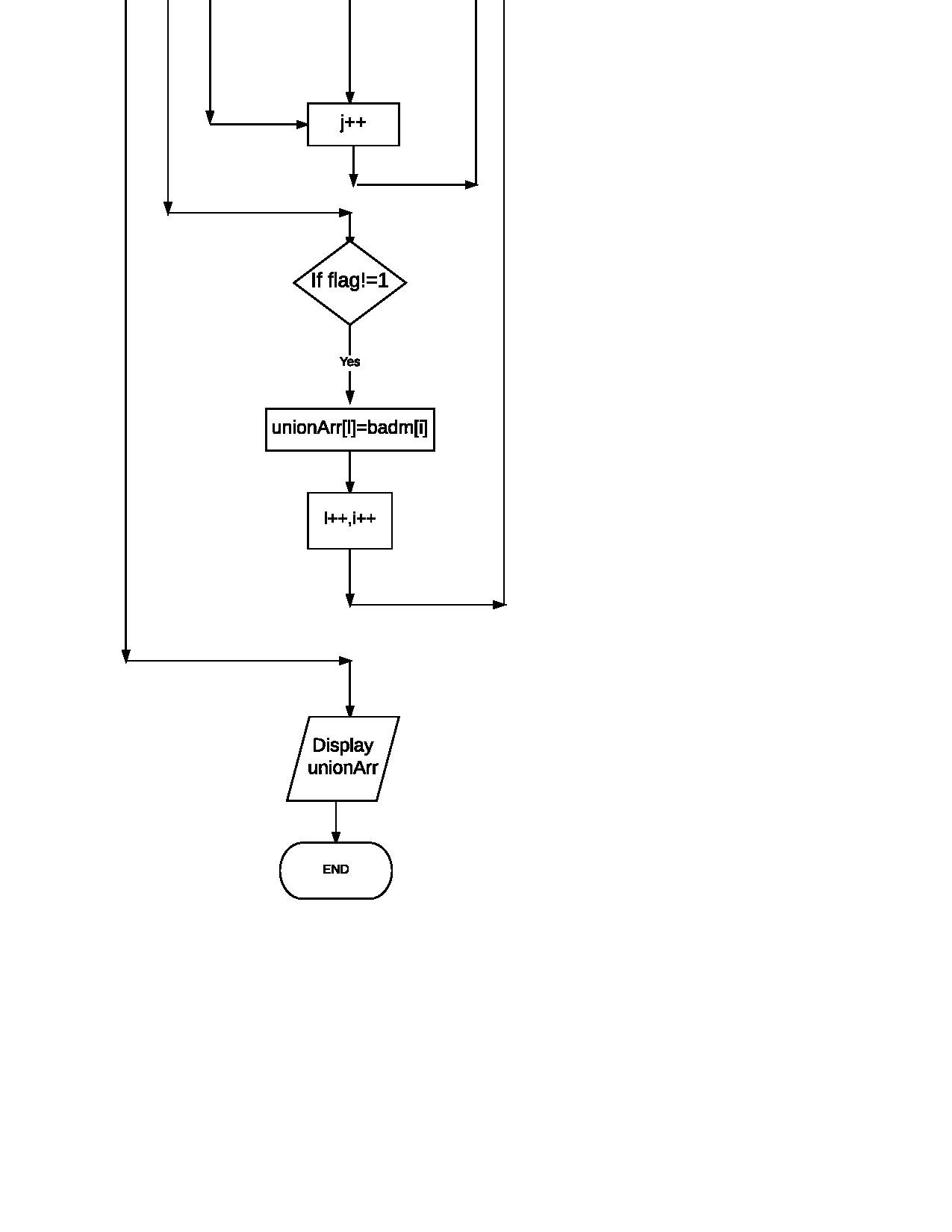
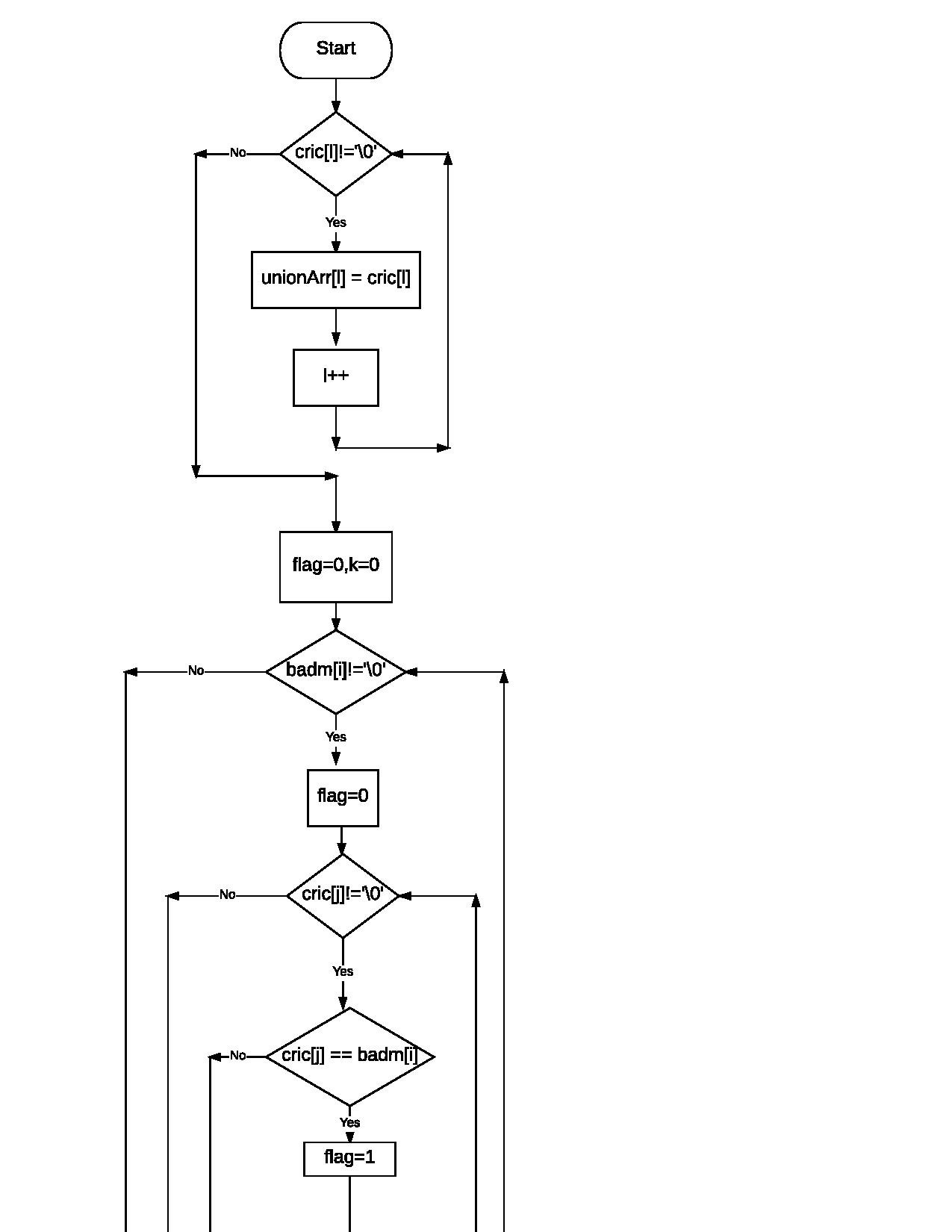
Store the value of currently set badm[] element into lyBadm[k]

Increment the value of k by 1

* 1. Increment the value of i by 1
  2. : Repeat the steps till the loop condition is satisfied

## Flowchart





Output:

