(All In All)

```
import java.util.Scanner;

public class Main {

	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);

		try
		{
			String str;
			while((str = sc.nextLine()) != null)
			{

				String[] s = str.split(" ");
				int j = 0;
				for(int i = 0; i < s[1].length(); i++)
				{

					if(str.charAt(j) == s[1].charAt(i))
						j++;

				}

				if(j == s[0].length())
					System.out.println("Yes");
				else System.out.println("No");
			}

		}
		catch(Exception e) {}

	}

}
```

(Sorting Algorithms)

```
/*
 * sorter.c
 *
 *  Created on: Sep 15, 2016
 *      Author: natha
 */

#include <stdio.h>
#include <stdlib.h>

void bubbleSort(int*, int);
void selectionSort(int*, int);
void insertionSort(int*, int);
void swap(int*, int, int, int);

int main()
{
	setbuf(stdout, NULL);

	int size;

	printf("Hello there user! This program is going to take an unsorted or sorted, if you're
feeling sporty, and sort it!\n");
	printf("Please specify how big you want the list: ");
	scanf(" %d", &size);

	int* bubbleList = (int*)calloc(size, sizeof(int));
	int* selectionList = (int*)calloc(size, sizeof(int));
	int* insertionList = (int*)calloc(size, sizeof(int));
	int value;

	printf("Please enter the values of of you're %d elements list: ", size);
	int i;
	for(i = 0; i < size; i++)
	{
		// Grabbing the input and creating the different lists to be sorted
		scanf(" %d", &value);
		bubbleList[i] = value;
		selectionList[i] = value;
		insertionList[i] = value;
	}
```

```
	for(i = 0; i < size; i++)
			printf("%d ", bubbleList[i]);
	printf("\n");

	// Calling each of the sorting methods
	printf("Bubble Sort:\n\n");
	bubbleSort(bubbleList, size);

	printf("The sorted list using the bubble sort method: ");
	for(i = 0; i < size; i++)
		printf("%d ", bubbleList[i]);


	printf("\n\nSelection Sort:\n\n");
	selectionSort(selectionList, size);

	printf("The sorted list using the selection sort method: ");
	for(i = 0; i < size; i++)
		printf("%d ", selectionList[i]);

	printf("\n\nInsertion Sort:\n\n");
	insertionSort(insertionList, size);

	printf("The sorted list using the insertion sort method: ");
		for(i = 0; i < size; i++)
			printf("%d ", insertionList[i]);

	// Freeing up the allocated memory for the lists
	free(bubbleList);
	free(selectionList);
	free(insertionList);

	printf("\nPress any key to continue...");
	getchar();
	getchar();
	return 0;
}

/** Takes a list and its size and sorts the list using the bubble method of sorting */
void bubbleSort(int* list, int size)
{
```

```
	int i;	// The marker used to keep track of how many times you've gone through the
array
	int j;	// The marker used to keep track of your comparisons
	for(i = 0; i < size - 1; i++)
		for(j = 0; j < size - 1; j++)
			if(list[j] > list[j + 1])
				swap(list, size, j, j + 1);
}

/** Takes a list and its size and sorts the list using the selection method of sorting */
void selectionSort(int* list, int size)
{
	int i;	// The marker used to keep track of how many times you've gone through the
array
	int j;	// The marker used to keep track of your comparisons
	for(i = 0; i < size; i++)
	{
		int currentMin = i;
		for(j = i; j < size - 1; j++)
			if(list[currentMin] > list[j + 1])
				currentMin = j + 1;

		if(i != currentMin)
			swap(list, size, i, currentMin);
	}
}

/** Takes a list and its size and sorts the list using the insertion method of sorting */
void insertionSort(int* list, int size)
{
	int in; // The inner "array" marker
	int out; // The outer "array" marker
	int insertion = 0; // The flag to see if an insertion happened or not
	for(out = 1; out < size; out++)
	{
		int tmp = list[out];
		in = out - 1;
		while(tmp < list[in] && in >= 0)
		{
			insertion = 1;
			list[out] = list[in];
			out--;
			in--;
```

```
                }
                list[in + 1] = tmp;
                if(insertion == 1)
                {
                        int i;
                        for(i = 0; i < size; i++)
                                printf("%d ", list[i]);
                        printf("\n");

                        insertion = 0;
                }
        }
}

/**
 * Takes a list, its size, and the two index locations in the list that needs to be swapped and
swaps them
 * as well as prints out the newly arranged list
 */
void swap(int* list, int size, int n, int m)
{
        int tmp = list[n];
        list[n] = list[m];
        list[m] = tmp;

        int i;
        for(i = 0; i < size; i++)
                printf("%d ", list[i]);
        printf("\n");
}
```

(Prime Words)

```
import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Scanner;

public class Main {
```

```
public static void main(String[] args)
{

	Scanner reader = new Scanner(System.in);
	ArrayList<String> input = new ArrayList<String>();
	int count = 0;

	int sums = 0;
	try
	{
	while(reader.hasNext())
	{
		input.add(reader.nextLine());
		count++;
	}
	}
	catch(Exception e) {}

	for(String lines : input)
	{
		boolean prime = true;
		for(Character chars : lines.toCharArray())
		{
			sums += (chars < 97) ? chars - 38 : chars - 96;
		}
		//System.out.println(sums);
		for(int i = 2; i < sums; i++)
		{

			if(sums % i == 0)
			{
				System.out.println("It is not a prime word.");
				prime = false;
				break;
			}
		}

		if(prime)
			System.out.println("It is a prime word.");
		sums = 0;
	}
```

```
        }

}
```