There are three pages. Please complete your answers in C (your choice of OS/compiler). Your solutions will be evaluated for:

1.	Correctness	Does your answer solve the problem?
2.	Clarity	Is your work easy to follow and understand? Adequate commenting is a plus.
3.	Formatting	Does your solution follow a consistent structure?
4.	Efficiency	Is your solution efficient with memory and processor time?
5.	Professionalism	Is your solution production-ready and does it meet professional standards?

1. Write a C function with the prototype:

```
int is_prime( const unsigned long X );
```

where `X` will be a positive integer less than 10,000,000.

`is_prime()` must return 0 if `X` is not a prime number and 1 if `X` is prime.

To test your implementation of `is_prime()`, write a program to verify your function by reading a set of sample numbers stored, one number per line, in a text file named `INPUT.TXT`. The program will print on a new line for each number:

A) The number read

and

B) "prime" or "NOT prime", as determined by your `is_prime()` function.

_____

Example INPUT.TXT:

```
49
4568923
3
9456872
```

Example output:

```
49 NOT prime
4568923 prime
3 prime
9456872 NOT prime
```

There are three pages. Please complete your answers in C (your choice of OS/compiler). Your solutions will be evaluated for:

1.	Correctness	Does your answer solve the problem?
2.	Clarity	Is your work easy to follow and understand? Adequate commenting is a plus.
3.	Formatting	Does your solution follow a consistent structure?
4.	Efficiency	Is your solution efficient with memory and processor time?
5.	Professionalism	Is your solution production-ready and does it meet professional standards?

2. Write a program that reads a string passed on the command line and prints the string with the words in reversed order.

   The string may be passed as one parameter or as several parameters; the output should be the same.

Example execution:

```
myprogram.exe This is the string to reverse
```

Example output:

```
reverse to string the is This
```

Example execution:

```
myprogram.exe “here is another string.”
```

Example output:

```
string. another is here
```

There are three pages. Please complete your answers in C (your choice of OS/compiler). Your solutions will be evaluated for:

1.	Correctness	Does your answer solve the problem?
2.	Clarity	Is your work easy to follow and understand? Adequate commenting is a plus.
3.	Formatting	Does your solution follow a consistent structure?
4.	Efficiency	Is your solution efficient with memory and processor time?
5.	Professionalism	Is your solution production-ready and does it meet professional standards?

3. Write a program that reads a list of integers from a text file and stores them in a binary tree.

   The numbers will be stored one number per line in a text file named `INPUT.TXT`.

   After reading and storing the last number, print the list of numbers, one per line, in order from smallest to largest.

```
Example INPUT.TXT:
      49
      4568923
      3
      9456872
      -405


Example output:
      -405
      3
      49
      4568923
      9456872
```