1.(1 pt)
Check ALL correct statments about the following algorithms.

(a) { **procedure** *double* (n: positive integer)
while n ¿ 0
begin
n := 2n
end
output(n) }

- A. This algorithm lacks finiteness.
- B. When n=5 is the input, n=10 is the final output of the algorithm.
- C. When n=5 is the input, the while loop is infinite.

(b) { **procedure** *divide* (n: positive integer)
while $n \geq 0$
begin
m := 1/n
n := n - 1
end
output(m) }

- A. This algorithm works and always outputs 1.
- B. This algorithm works and outputs 1/n.
- C. This algorithm lacks definateness since division by zero occurs.
- D. When n=1 is the input, on the second iteration of the while loop a division by zero occurs.
- E. When n=1 is the input, the algorithm exits the while loop after the first iteration and outputs m=1.
- F. When n=1 is the input, after the first iteration of the while loop we have m=1 and n=0.

(c) { **procedure** *sum* (n: positive integer)
sum := 0
while i ¡ 10
begin
sum := *sum* + i
end
output(*sum*) }

- A. This algorithm does not seem to use the input n.
- B. If i is initialized to the value 1 in the begining of the algorithm, the algorithm works and outputs 1.
- C. If i is initialized to the value 1 in the begining of the algorithm, the algorithm is still not finite since it gets stuck in an infinite while loop.
- D. This algorithm lacks preciseness since the value of i is not initialized.
- E. If i is initialized to the value 1 in the begining of the algorithm, the algorithm works and outputs $1+2+\cdots+9$.
- F. If i is initialized to the value 1 in the begining of the algorithm, the algorithm works and outputs $1+2+\cdots+10$.

2.(1 pt) This exercise refers to the binary search algorithm given below.

procedure *binary search* (x : integer, $a_1, a_2, ..., a_n$: increasing integers)
i := 1 { i is left endpoint of search interval }
j := n { j is right endpoint of search interval }
while $i < j$
begin
$m := \lfloor (i+j)/2 \rfloor$
if $x > a_m$ **then** $i := m + 1$
else $j := m$
end
if $x = a_i$ **then** *location* := i
else *location* := 0
{ *location* is the subscript of term equal to x , or 0 if x is not found }

Suppose our list of increasing integers is shown in the table below

8	1	4	9	13	12	15	5

Suppose we conduct the binary search algorithm on this list where we search for 15.

(a) In the language of the algorithm above, enter the correct values for the following variables for this particular search:

$x =$ _____ $n =$ _____ $a_2 =$ _____ $a_4 =$ _____

(b) After the first iteration of the while loop, what are the values of the following variables?

$i =$ _____ $j =$ _____ $m =$ _____

(c) Intuitively, after the first iteration of the while loop, we have cut down our search to a set S of roughly half of the original numbers on the list. Check the numbers that are in the set S after the first iteration of the while loop:
S has the following elements:

- A. 9
- B. 5
- C. 8
- D. 1
- E. 13
- F. 15
- G. 4
- H. 12

(d) After the second iteration of the while loop, some of the variables have altered values. Enter the values of the following variables after the second while loop iteration:

$i =$ _____ $j =$ _____ $m =$ _____

(e) After the third iteration of the while loop, some of the variables have altered values. Enter the values of the following variables after the third while loop iteration:

$i =$ _____ $j =$ _____ $m =$ _____

(f) After the 3rd iteration of the while loop, the while loop terminates and the variable 'location' is computed. Enter the value for this variable in this case:

location = _____

3.(1 pt) There are many methods to calculate

$$x^{2^k}$$

for some positive integer k and given x.

The first DIRECT METHOD, just multiplies x with itself to find x^2 and then multiplies x with x^2 to find x^3 and keeps going like this till x^{2^k} is found.

Thus for example to find x^4 it requires 3 multiplications: (x with itself to find x^2, x with x^2 to find x^3 and finally x with x^3 to find x^4.)

On the other hand there is also a second method, the SQUARING METHOD, where we at each stage multiply the last number with itself.

For example to find x^4, one first multiplies x with itself to get x^2, then one multiplies x^2 with itself to get x^4.

Notice this second method only took 2 multiplications to find x^4.

Answer the following questions:

To calculate x^{16} with the first direct method takes _____ multiplications.

To calculate x^{16} with the second, squaring method takes _____ multiplications.

Which method is more efficient for large k?

- A. The DIRECT METHOD is more efficient than the SQUARING METHOD.
- B. The SQUARING METHOD is more efficient than the DIRECT METHOD.

4.(1 pt) You are the head of a division of a big Silicon Valley company and have assigned one of your engineers, Jim, the job of devising an algorithm to sort through an English text of n words and convert it into an Esperanto document. Jim comes up with an algorithm which takes $2n^2 + 2^n$ bit operations to handle an input text with n words. Suppose the computers in your business can handle one bit operation every nanosecond (1 nanosecond $= 10^{-9}$ seconds).

(a) How many nanoseconds would it take Jim's algorithm to convert a text with 13 words on these computers?

(b) How many HOURS would it take Jim's algorithm to convert a text with 56 words on these computers?(Do not round your answers for WeBWorK.) _____

(Recall a million is 10^6, a billion is 10^9 and a trillion is 10^{12}.)

For an input text of 100 words, the statement that best describes the performance of Jim's algorithm is:

- A. His algorithm would take between 40 and 400 years to run.
- B. His algorithm would take between 40 million and 40 billion years to run.
- C. His algorithm would take between 40 billion and 40 trillion years to run.
- D. His algorithm would take between 4000 and $40,000$ years to run.
- E. His algorithm would take between 4 and 40 years to run.
- F. His algorithm would take more than 40 trillion years to run.
- G. His algorithm would take between 40 thousand and 40 million years to run.