2. Refer to your Personalized Project Reference when answering this question.

A. Consider the first iteration statement in the Procedure section of your Personalized Project Reference. Identify the variable(s) that determine when the iteration statement stops iterating. Identify specific values for the variable(s) that will cause the iteration statement to stop. Explain why these value(s) would cause the iteration statement to stop. If there are no such variable(s) or specific value(s), explain how the iteration statement stops.

B. Consider the procedure included in part (i) of the Procedure section of your Personalized Project Reference. Another programmer is planning to use your procedure in a different program and wants to write some test cases to check whether it will work. Write a procedure call with specific argument(s) that your procedure will accept but will cause your procedure to behave incorrectly. Describe the incorrect behavior. Explain why the incorrect behavior occurs as a result of this call. If it is not possible for a call with accepted arguments to cause your procedure to behave incorrectly, explain why this is the case for your procedure.

C. Consider the list included in part (i) of the List section of your Personalized Project Reference. Explain how the list uses abstraction to manage complexity in your program. Suppose the list was not included in your program. Describe how you would need to adjust the code segment in part (ii) of the List section of your Personalized Project Reference so that it has the same behavior. If it is not possible to have the same behavior without the use of the list, explain why.

STOP
END OF EXAM

2. Refer to your Personalized Project Reference when answering this question.

(a) Consider the first iteration statement included in the Procedure section of your Personalized Project Reference. Describe what is being accomplished by the code in the body of the iteration statement.

(b) Consider the procedure identified in part (i) of the Procedure section of your Personalized Project Reference. Write two calls to your procedure that each cause a different code segment in the procedure to execute. Describe the expected behavior of each call. If it is not possible for two calls to your procedure to cause different code segments to execute, explain why this is the case for your procedure.

(c) Suppose another programmer provides you with a procedure called `checkValidity(value)` that returns `true` if a value passed as an argument is considered valid by the other programmer and returns `false` otherwise. Using the list identified in the List section of your Personalized Project Reference, explain in detailed steps an algorithm that uses `checkValidity` to check whether all elements in your list are considered valid by the other programmer. Your explanation must be detailed enough for someone else to write the program code for the algorithm that uses `checkValidity`.

Write your responses to this question only on the designated pages in the separate Written Response booklet. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

STOP

END OF EXAM

2. Refer to your Personalized Project Reference when answering this question.

   (a) Consider the first conditional statement included in the Procedure section of your Personalized Project Reference. Describe your conditional statement, including its Boolean expression. Describe what the procedure does in general when the Boolean expression of this conditional statement evaluates to `false`.

   (b) Consider the procedure and procedure call identified in parts (i) and (ii) of the Procedure section of your Personalized Project Reference. Describe the outcome that your procedure call is intended to produce. Write a new procedure call with at least one different argument value that will produce the same outcome, if possible, and explain why this procedure call produces the same outcome. If it is not possible to write a new procedure call that produces the same outcome, explain why this is not possible.

   (c) Consider the procedure identified in part (i) of the Procedure section of your Personalized Project Reference. Identify the parameter(s) used in this procedure. Explain how your identified parameter(s) use abstraction to manage complexity in your program.

Write your responses to this question only on the designated pages in the separate Written Response booklet. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

STOP

END OF EXAM