



# Fundamental Competence of App Inventor Programming

## MIT App Inventor 2 Examination Paper

### **Exam instructions**

1. This exam is a practical exam. Maximum time allowed for this exam is 120 minutes. Please finish all the requirements with the exam time. The full mark means 100 points and you will pass the exam with at least 70-points.
2. There are two questions of this practical exam. Each question count 50 points. The total score is 100 points.
3. All the necessary files are located in respective folders under C:\ANS.CSF\. Please follow the instructions to save your files in respective folders under C:\ANS.CSF\ and close the MIT App Inventor IDE.
4. Please notice that the markings of each hands-on question are related to each other. Components' properties or other settings which are not clearly asked by the instruction are kept as its original state.
5. While reading the question instructions, please notice that "0" means the number zero and "O" means the corresponding alphabet. The mouse is set as the right-handed user. Mouse button settings are for right-handed.
6. If you have any questions, please raise your hand and ask questions. Do not talk privately.

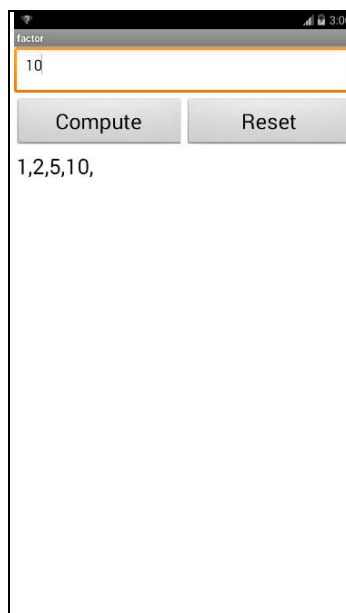
- Practical exam 100% (Each question count 50 points.)

## 1. Finding all the factors

### A. Question description:

Please open the **MID01.aia** in your App Inventor IDE and finish this app. Input a number in the Textbox and all the factors of this input number will be shown when the “**Compute**” button is pressed. Another prompt for the error messages will show up if the input is incorrect. Please follow the instructions below to finish this app and save the file as **MID01.apk**. Finally, please download the .aia and the .apk file of this question and save them in the corresponding folder under C:\ANS.CSF.

*Note: all the necessary components are pre-added in the MID01.aia.*



< Illustrative layout >

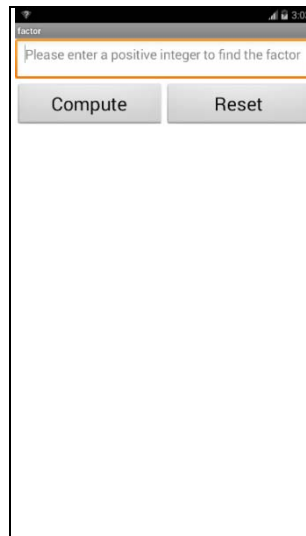
### B. Program specs:

- (1) As in the **Designer** page, all the necessary components of this question have been provided, please refer to the image below to finish the layout.
- (2) The two buttons under the **TextBox** component have to be arranged horizontally and fill the whole screen width. Each button's width properties have to be set to 50%.
- (3) When the “**Compute**” button is pressed; all the factors of the input number will be displayed on the **Label** component. The term “**All the factors**” means numbers including 1, the number itself and all possible factors between. Finally, pressed the “**Reset**” button to input another number.

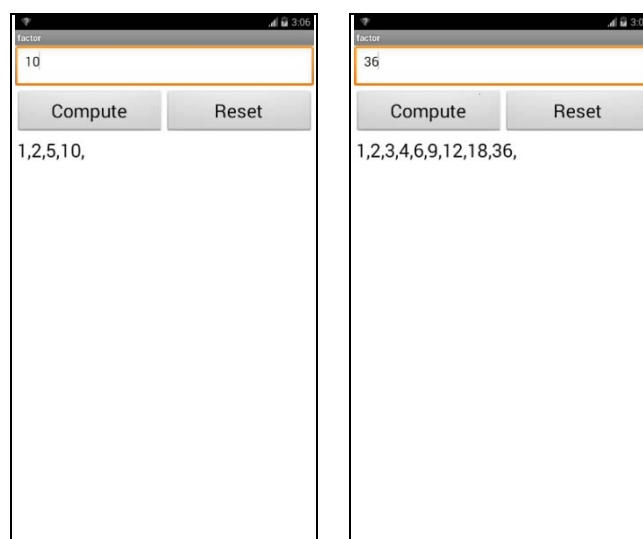
- (4) A prompt will show with the message “**Empty input, decimal numbers and numbers less than or equal to zero are NOT allowed.**” if the “**Compute**” button is pressed when (1) The **TextBox** components has no input at all. (2) The input number is less than or equal to zero. (3) The input number is a decimal number. Finally, clear the text properties of the **TextBox** and the **Label** components.

C. Example screenshots:

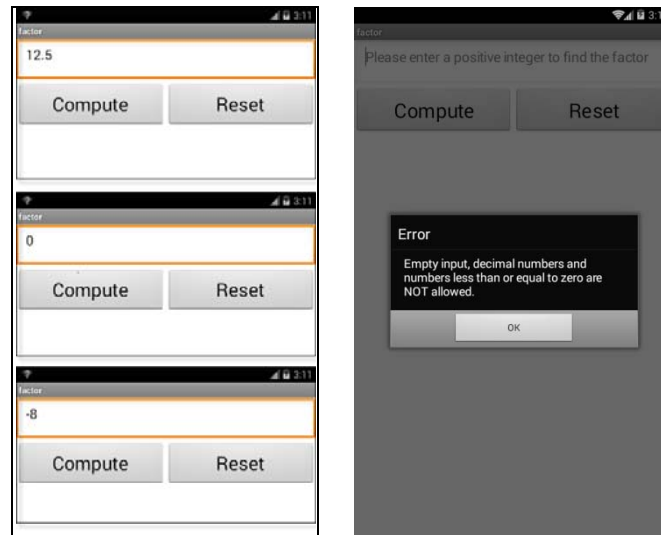
- (1) Start/Initial screen.



- (2) When the “**Compute**” button is pressed; all the factors of the input number will be displayed on the **Label** component. The term “**All the factors**” means numbers including 1, the number itself and all possible factors between. Finally, pressed the “**Reset**” button to input another number.



- (3) A prompt will show with the message “**Empty input, decimal numbers and numbers less than or equal to zero are NOT allowed.**” if the “**Compute**” button is pressed when (1) The **TextBox** components has no input at all. (2) The input number is less than or equal to zero. (3) The input number is a decimal number. Finally, clear the text properties of the **TextBox** and the **Label** components.



D. Test your app:

- (1) The layout is identical (or at least similar to the reference layout). All components' properties are set correctly.
- (2) All the factors of the input number are displayed on the **Label** components after the “**Compute**” button is pressed.
- (3) A prompt will show with the message “**Empty input, decimal numbers and numbers less than or equal to zero are NOT allowed.**” if the “**Compute**” button is pressed when (1) The **TextBox** components has no input at all. (2) The input number is less than or equal to zero. (3) The input number is a decimal number. Finally, clear the text properties of the **TextBox** and the **Label** components.

E. Assessment Criteria:

Criteria	Points	Score
(1) The layout is identical (or at least similar to the reference layout. All components' properties are set correctly.	5	
(2) All the factors of the input number are displayed on the <b>Label</b> components after the " <b>Compute</b> " button is pressed.	30	
(3) A prompt will show with the message " <b>Empty input, decimal numbers and numbers less than or equal to zero are NOT allowed.</b> " if the " <b>Compute</b> " button is pressed when (1) The <b>TextBox</b> components has no input at all. (2) The input number is less than or equal to zero. (3) The input number is a decimal number. Finally, clear the text properties of the <b>TextBox</b> and the <b>Label</b> components.	15	
Total Score	50	

## 2. Multiplication Challenge

### A. Question description:

Please open the **MID02.aia** in your App Inventor IDE and finish this app. After the app is started, there will be a calculation of two numbers' multiplication. Enter the answer and press the “**Submit**” button, the app will tell you whether you have input the right answer. Please follow the instructions below to finish this app and save the file as **MID02.apk**. Finally, please download the .aia and the .apk file of this question and save them in the corresponding folder under C:\ANS.CSF.

*Note: all the necessary components are pre-added in the MID02.aia.*



< Illustrative layout >

### B. Program specs:

- (1) The label (NO1) is the multiplicand and the label (NO2) is the multiplier. These two numbers will be randomly assigned within the range from 2 to 9. The user has to enter the multiplication result in the **TextBox** component.
- (2) The **TextBox** component must be set as “**NumbersOnly**”.
- (3) The app will check whether the input number is correct. If the answer is correct, the total times of the correct-answered questions will be accumulated and show a message “**correct**” on the **Label** component; if not, the total times of the correct-answered questions will not be accumulated and show a message “**wrong**” on the **Label** component.
- (4) Whether the answer is correct or not, the **TextBox** component will be reset to blank and be ready for the next input after the “**Submit**” button is pressed.

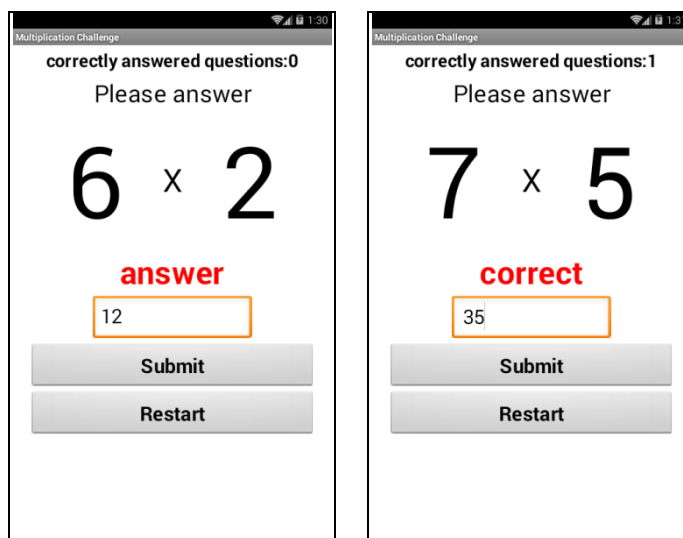
- (5) When the “**Restart**” button is pressed, the total times of the correct-answered questions will be reset to zero; the Text property of the result **Label** component and the **TextBox** component will be reset to blank for the next input.

C. Example screenshots:

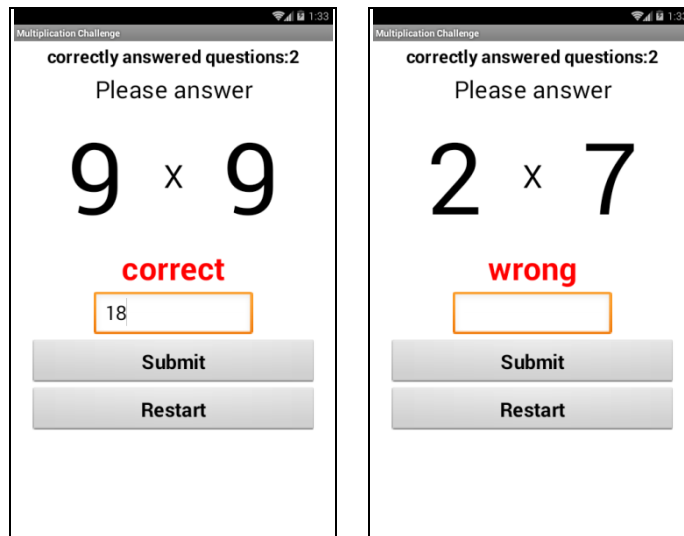
- (1) Start/Initial screen.



- (2) The app will decide whether the input number is correct. If the answer is correct, the total times of the correct-answered questions will be accumulated and show a message “**correct**” on the **Label** component and reset the Text property of the **TextBox** to blank.



- (3) If the answer is incorrect, the total times of the correct-answered questions will not be accumulated and show a message “**wrong**” on the **Label** component.



- (4) When the “**Restart**” button is pressed, the total times of the correct-answered questions will be reset to zero, the **Text** property of the result Label component and the **TextBox** component will be reset to blank for the next input.



#### D. Test your app:

- (1) The layout is identical (or at least similar) to the reference layout. All components' properties are set correctly.
- (2) The **TextBox** component must be set as “**NumbersOnly**”.
- (3) After the app is started, the multiplicand and the multiplier have to be randomly assigned within the range from 2 to 9.



- (4) After user input the answer, the total times of the correct-answered questions will be accumulated and show a message “**correct**” on the **Label** component if the answer is correct.
- (5) If the answer is not correct, the total times of the correct-answered questions will not be accumulated and show a message “**wrong**” on the **Label** component.
- (6) When the “**Restart**” button is pressed, the total times of the correct-answered questions will be reset to zero, the Text property of the result **Label** component and the **TextBox** component will be reset to blank for the next input.

E. Assessment Criteria :

Criteria	Points	Score
(1) The layout is identical (or at least similar) to the reference layout.	6	
(2) The <b>TextBox</b> component must be set as “ <b>NumbersOnly</b> ”.	5	
(3) After the app is started, the multiplicand and the multiplier have to be randomly assigned within the range from 2 to 9.	8	
(4) After user input the answer, the total times of the correct-answered questions will be accumulated and show a message “ <b>correct</b> ” on the <b>Label</b> component if the answer is correct.	15	
(5) If the answer is not correct, the total times of the correct-answered questions will not be accumulated and show a message “ <b>wrong</b> ” on the <b>Label</b> component.	8	
(6) When the “ <b>Restart</b> ” button is pressed, the total times of the correct-answered questions will be reset to zero, the Text property of the result <b>Label</b> component and the <b>TextBox</b> component will be reset to blank for the next input.	8	
Total Score	50	