Question 5 **[15 marks]**

"Math Quiz" is a game of combining digits with mathematics operators to form an equation in order to produce the target number. The digits (1-9) and mathematics operators (+, –, ×) are arranged with GridLayout. While selecting the digits and operators, the intermediate equation is shown in the TextView with id "equation". After completed the equation, the player can check the answer by clicking the button with id "buttoncheck". If the resulted value from the suggested equation matched with the desired one, a new question will be generated. Otherwise, notification will be shown with an alert dialog.

Listing below depicts the layout resource file "activity_main.xml".

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Target Value: " />

    <GridLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:columnCount="3" >

        <Button
            android:id="@+id/button1"
            android:text="1" />
        <Button
            android:id="@+id/button2"
            android:text="2" />
        <Button
            android:id="@+id/button3"
            android:text="3" />
        <Button
            android:id="@+id/button4"
            android:text="4" />
        <Button
            android:id="@+id/button5"
            android:text="5" />
        <Button
            android:id="@+id/button6"
            android:text="6" />
        <Button
            android:id="@+id/button7"
            android:text="7" />
        <Button
            android:id="@+id/button8"
            android:text="8" />
        <Button
            android:id="@+id/button9"
            android:text="9" />
        <Button
            android:id="@+id/buttonsummation"
            android:text="+" />
        <Button
            android:id="@+id/buttonsubtraction"
            android:text="-" />
        <Button
            android:id="@+id/buttonmultiplication"
            android:text="x" />
    </GridLayout>

    <TextView
        android:id="@+id/equation"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Typed Equation: " />

    <Button
        android:id="@+id/buttoncheck"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Check" />

</LinearLayout>
```

Listing below depicts part of the activity class “MainActivity.java”.

```
public class MainActivity extends Activity implements OnClickListener {
    private Button button [] = new Button[9];
    private Button buttonSummation, buttonSubtraction, buttonMultiplication, buttonCheck;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button[0] = (Button)findViewById(R.id.button1);
        button[1] = (Button)findViewById(R.id.button2);
        button[2] = (Button)findViewById(R.id.button3);
        button[3] = (Button)findViewById(R.id.button4);
        button[4] = (Button)findViewById(R.id.button5);
        button[5] = (Button)findViewById(R.id.button6);
        button[6] = (Button)findViewById(R.id.button7);
        button[7] = (Button)findViewById(R.id.button8);
        button[8] = (Button)findViewById(R.id.button9);

        for (int i=0; i<button.length; i++)
            button[i].setOnClickListener(this);

        buttonSummation = (Button)findViewById(R.id.buttonsummation);
        buttonSummation.setOnClickListener(this);
        buttonSubtraction = (Button)findViewById(R.id.buttonsubtraction);
        buttonSubtraction.setOnClickListener(this);
        buttonMultiplication = (Button)findViewById(R.id.buttonmultiplication);
        buttonMultiplication.setOnClickListener(this);

        buttonCheck = (Button)findViewById(R.id.buttoncheck);
        buttonCheck.setOnClickListener(this);
    }

    // Button click event handling
    public void onClick(View v) {
        // Add your code here
    }
}
```

(a) **Illustrate** with diagram about what you would see when the Android application program “MainActivity” of the “Math Quiz” game is started.

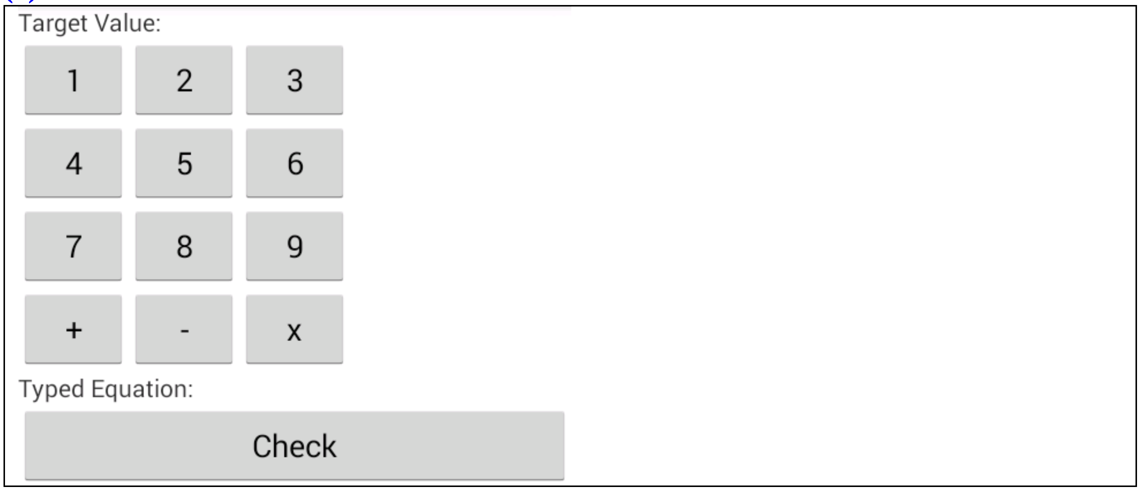
[5]

(b) **Complete** the button clicked event handler method “public void onClick(View v)” in pseudo code.

[10]

Question 5 **[15 marks]**

(a)



(b)

```
Get player selected value from the clicked button
If (digit or mathematics operator)
    -   Concatenate with the typed equation
    -   Update the TextView "equation"
If (check button)
    -   Evaluate if the equation can produce the target value
    -   If (match)
            o   Generate a new question
            o   Show next question  (or Update the display)
    -   Otherwise
            o   Notify the player with an alert dialog
```

Question 5 **[17 marks]**

“Find LCM” is a game of find the LCM (least common multiple) of two numbers. LCM is the smallest positive number which is divisible by two integers. For example, 12 is the LCM of 3 and 4. In the game “Find LCM”, four positive integers are placed within the game board and arranged with TableLayout. Only one of the numbers is the LCM. The player needs to pick up the LCM in the shortest time. For each game, five questions are needed to be answered correctly. After the game completed, the total time used will be shown with an alert dialog.

Listing below depicts the layout resource file “activity_main.xml”.

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/atextview"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Select the LCM of 8 and 7"/>

    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:stretchColumns="*" >

        <TableRow>
            <Button
                android:id="@+id/button1"
                android:text="56"/>
            <Button
                android:id="@+id/button2"
                android:text="14"/>
        </TableRow>

        <TableRow>
            <Button
                android:id="@+id/button3"
                android:text="48"/>
            <Button
                android:id="@+id/button4"
                android:text="24"/>
        </TableRow>

    </TableLayout>
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/startbutton"
        android:text="Start game"
        />

</LinearLayout>
```

Listing below depicts part of the activity class "MainActivity.java".

```
public class MainActivity extends Activity implements OnClickListener {
...
    private Button button1;
    private Button button2;
    private Button button3;
    private Button button4;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button1 = (Button)findViewById(R.id.button1);
        button1.setOnClickListener(this);
        button2 = (Button)findViewById(R.id.button2);
        button2.setOnClickListener(this);
        button3 = (Button)findViewById(R.id.button3);
        button3.setOnClickListener(this);
        button4 = (Button)findViewById(R.id.button4);
        button4.setOnClickListener(this);
...
    }

    // Button click event handling
    public void onClick(View v) {
        // Add your code here
    }
}
```

(a) **Illustrate** with diagrams about what you would see when the Android application program "MainActivity" of the "Find LCM" game is started.

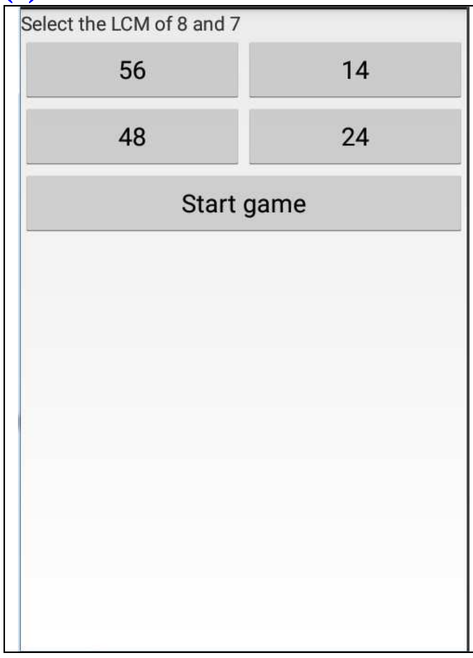
[5]

(b) **Complete** the event handler method "public void onClick(View v)" in pseudo code.

[12]

Question 5 **[17 marks]**

(a)



Select the LCM of 8 and 7

(b)

```
- Get player selected value from the clicked button
- If it is the correct answer (i.e., the LCM is selected)
    o Increment number of trials
    o If game ends (game ends if total number of correct trials is 5)
        ▪ Reset the game board
        ▪ Evaluate the time consumed
        ▪ Notify the player with an alert dialog
    o Otherwise
        ▪ Generate and show next question
```

Question 5 **[13 marks]**

“Fruit Match” is a game for children to recognize the pronunciation of some common fruits. The application first pronounces the name of a fruit which is randomly selected. The child then listens to the pronunciation and tries to depict the corresponding fruit image. There are four fruit images displayed in the user interface and arranged with TableLayout. If the child wants to listen to the pronunciation again, he/she can click the “Pronounce” button. After the child selected an image, it will check if it matches with the pronounced fruit type. If so, a circle (“O”) will be drawn on the selected image. Otherwise, a cross (“X”) will be sketched. When he/she clicks the “Next Fruit” button, a new fruit type is randomly selected and pronounced while another set of fruit images will also be generated randomly for the player selection.

Listing below depicts the layout resource file “activity_main.xml”.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/atextview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Fruit Match"/>

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:stretchColumns="*" >

        <TableRow>
            <ImageButton
                android:id="@+id/button1"
                android:src="@drawable/apple"/>
            <ImageButton
                android:id="@+id/button2"
                android:src="@drawable/banana"/>
        </TableRow>

        <TableRow>
            <ImageButton
                android:id="@+id/button3"
                android:src="@drawable/orange"/>
            <ImageButton
                android:id="@+id/button4"
                android:src="@drawable/pear"/>
```

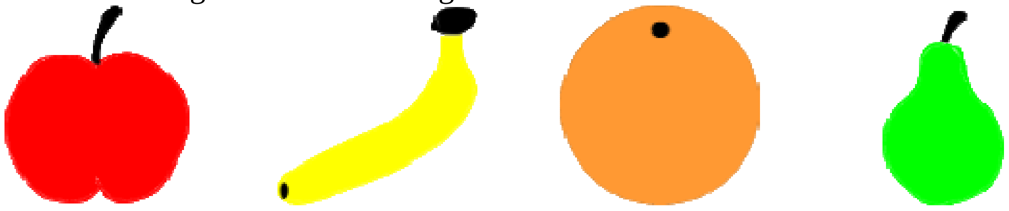
```
      </TableRow>
   </TableLayout>

   <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Hint: Click the 'Pronounce' button to listen again" />

   <Button
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@+id/pronouncebutton"
      android:text="Pronounce"
      />

   <Button
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@+id/nextbutton"
      android:text="Next Fruit"
      />

</LinearLayout>
```

Listing below depicts part of the activity class “MainActivity.java”.

```
public class MainActivity extends Activity implements OnClickListener {
   private static int [] imageButtonId = { R.id.button1, R.id.button2,
                                            R.id.button3, R.id.button4};
   private ImageButton imageButton [] = new ImageButton[4];
   private Button startButton, pronounceButton;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      startButton = (Button)findViewById(R.id.nextbutton);
      startButton.setOnClickListener(this);

      pronounceButton = (Button)findViewById(R.id.pronouncebutton);
      pronounceButton.setOnClickListener(this);

      for (int i=0; i<imageButton.length; i++) {
         imageButton[i] = (ImageButton)findViewById(imageButtonId[i]);
         imageButton[i].setOnClickListener(this);
      }
…
```

```
    }

    // Button click event handling
    public void onClick(View v) {
        // Add your code here
…
    }
…
}
```

Different fruit images are shown in Figure 5.



(a) "drawable/apple" (b) "drawable/banana" (c) "drawable/orange" (d) "drawable/pear"

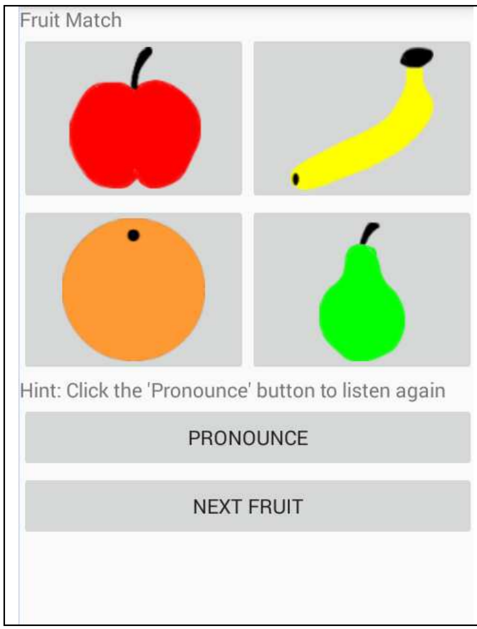
Figure 5. Fruit images.

(a) **Illustrate** with diagram about what you would see when the Android application program "MainActivity" of the "Fruit Match" game is started.

[5]

(b) **Complete** the event handler method "public void onClick(View v)" in pseudo code style.

[8]



(b)

```
If pronounce button is clicked
    -   Play the fruit type sound file
If start button is clicked
    -   Random select another fruit type
    -   Random generate another set of fruit images
Else
    -   Evaluate if correct fruit image is selected
            o   Draw a circle (“O”) on the selected image
    -   Otherwise
            o   Draw a cross (“X”) on the selected image
```

Question 5 **[17 marks]**

In the Shape Quiz game, the game player is required to answer five questions for testing their knowledge about basic shapes (circle, rectangle, square, and triangle). After five questions have been completed, an alert dialog will be shown to notify the player his/her score.

Listing below depicts the layout resource file "main.xml".

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="vertical" >

   <TextView
      android:id="@+id/statustextview"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Welcome to Shape Quiz!" />

   <ImageView
      android:id="@+id/imageview1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center"
      android:src="@drawable/ready" />

   <RadioGroup
      android:id="@+id/radiogroup1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Your choice"
      >

      <RadioButton android:id="@+id/choice1"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Circle" />

      <RadioButton android:id="@+id/choice2"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Rectangle" />

      <RadioButton android:id="@+id/choice3"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Square" />
```

```
        <RadioButton android:id="@+id/choice4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Triangle" />

    </RadioGroup>

    <Button
        android:id="@+id/answerbutton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Next question" />

</LinearLayout>
```

Listing below depicts part of the activity class “ShapeQuizActivity.java”.

```
public class ShapeQuizActivity extends Activity {
    Button button; // Next question button
…
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
…
        button = (Button) findViewById(R.id.answerbutton);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            // Button click handling
            public void onClick(View v) {
                // Add your code here
            }
        });
    }
…
}
```

(a) **Illustrate** with diagram about what you would see when the Android application program "ShapeQuizActivity" of the Shape Quiz game is started. Figure 1 sketches the image resource "res/drawable/ready.jpg".

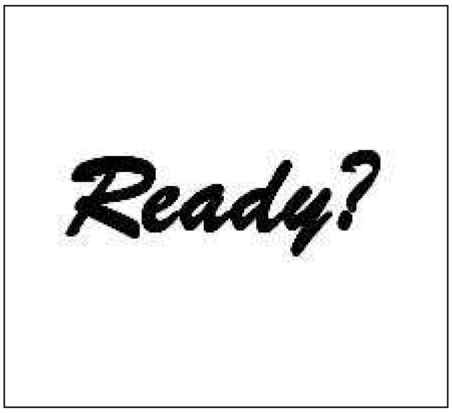


Figure 1. Image resource "res/drawable/ready.jpg".

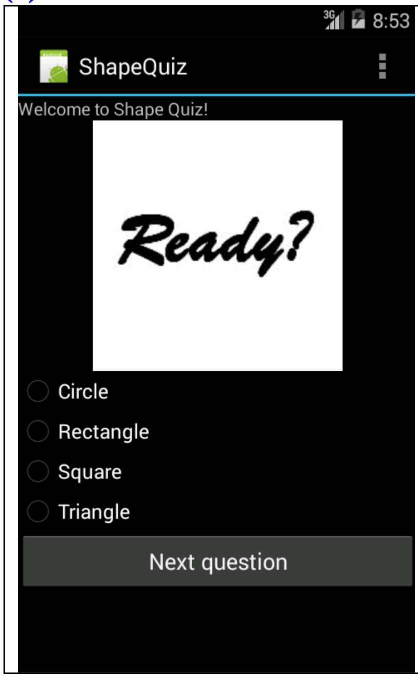
[6]

(b) For the Button object "button", it is clicked if the player proceeds to the next question. **Complete** the button clicked event handler method "public void onClick(View v)" in pseudo code.

[11]

Question 5 **[17 marks]**

(a)

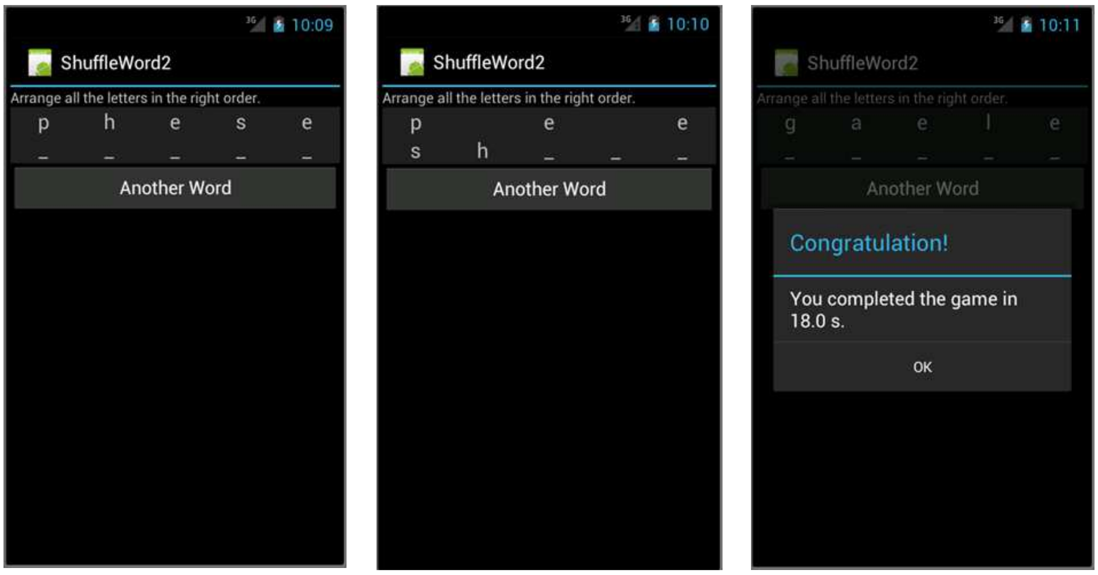


(b)

```
Input validation (determine if player selected a radio button)
If (one of the radio button is selected)
    -   Get result from the selected radio button (get player's selection)
    -   Check player's answer
            o   If (correct)
                    ▪   Accumulate the number of question answered correctly
    -   Increment the total number of question answered
    -   Determine if game ends (game ends if player answered all the five questions)
            o   If (game end)
                    ▪   Notify the player with an alert dialog
                    ▪   Reset user interface / End game
            o   Otherwise
                    ▪   Clear player selection in radio group
                    ▪   Randomize a shape of next question (generate next question)
Draw the corresponding shape image of next question
```

Question 5 **[15 marks]**

"ShuffleWord" is a game of arranging all the letters in the right order. According to Figure 5 (a), letters from a secret word are shuffled and displayed with GridView. In Figure 5 (b), each letter selected from the Griview will be shown in another Gridview which is located below it. If the player completes the game, an alert dialog will be popup to show the total time elapsed, which is illustrated in Figure 5 (c). The button "Another Word" provides the function of restarting a new game.



Congratulation!

You completed the game in 18.0 s.

(a) Shuffled letters are sketched in the first GridView.

(b) After letters "f" and "i" are selected by player.

(c) An alert dialog is popup if the player completes the game.

Figure 5. Game "ShuffleWord"

Listing below depicts part of the layout resource file "main.xml"

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
…
>
  …
</LinearLayout>
```

Listing below depicts part of the activity class "ShuffleWordActivity.java"

```
public class ShuffleWordActivity extends Activity implements OnClickListener {
  // Instance of ShuffleWord
  private ShuffleWord shuffleWord;
  // Recording the start time
  private long startTime = 0;
```

```
    // The adapter of the grid of shuffled word
    private ArrayAdapter<String> wordAdapter;
    // The adapter of the grid of player selection
    private ArrayAdapter<String> playerAdapter;
    // The grid of shuffled word
    private GridView wordGridView;
    // The grid of player selection
    private GridView playerGridView;
    // New game button
    private Button newGameButton;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        newGameButton = (Button)findViewById(R.id.newgamebutton);
        newGameButton.setOnClickListener(this);

        shuffleWord = new ShuffleWord(this);

        wordGridView = (GridView) findViewById(R.id.wordgridview);
        playerGridView = (GridView) findViewById(R.id.playergridview);
…
        wordGridView.setOnItemClickListener(
            new AdapterView.OnItemClickListener() {
                @Override
                public void onItemClick(AdapterView<?> parent, View v,
                                        int position, long id) {
…
                }
        });

        startTime = System.currentTimeMillis();    // Record start time
    }

    // Player click one of the three buttons
    public void onClick(View v) { … }

    // Restart of the game
    private void resetGame() { … }

    // Show statistics after completion of the game
    private void gameCompleted() { … }
}
```

(a) **Complete** the layout resource file "main.xml" of the above Android game. Four child elements are enclosed in the LinearLayout. The first widget is a TextView for displaying message "Arrange all the letters in the right order.". The second and third one are GridViews for showing the shuffled and selected letters respectively. The last widget is a buttons labeled with "Another Game".

[4]

(b) Shuffled letters are put into the items of the GridView. **Write** codes to assign data items get from instance “shuffleWord” to GridView object “wordGridView”.

It is given that resource file “\res\layout\item_view.xml” has been defined for the layout of items in the GridView. Data items can be obtained with method “public ArrayList<String> getAnswerWordList()” provided by class “ShuffleWord”.

[3]

(c) For the first GridView object "wordGridView”, item is selected by the player for arranging the letter in correct sequence. **Complete** the event handler method “public void onItemClick(AdapterView<?> parent, View v, int position, long id)” of the GridView object “wordGridView” in pseudo code.

[8]

[End of Question 5]

Part II [45 marks]

Question 5 **[15 marks]**

(a)

```
<TextView
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="@string/game_msg"
/>

<GridView
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:id="@+id/wordgridview"
/>

<GridView
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:id="@+id/playergridview"
/>

<Button
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:id="@+id/newgamebutton"
   android:text="Another Word"
/>
```

(b)

```
wordAdapter = new ArrayAdapter<String>(this,
                                       R.layout.item_view,
                                       shuffleWord.getAnswerWordList());
wordGridView.setAdapter(wordAdapter);
```

(c)

```
Get the selected value from the clicked item
Validate player selected value (i.e., check if the label of clicked item is empty or not)
If (valid input)
    -  Update the selected item of the first GridView (empty the selected item)
    -  Update the selected item of the second GridView (filled with the selected letter)
    -  Determine if game ends (game ends if player arranged all the letters in specific
       order)
        o  If (game end)
            ▪  Evaluate the time consumed
            ▪  Notify the player with an alert dialog
Update the two grid displays
```

[End of Question 5]