

```

public class FirstActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        .....
        this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        .....
        Button buttoplay = (Button) findViewById(R.id.buttonplay);
        buttoplay.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    startActivity(new Intent(getApplicationContext(),ThirdActivity.class));
                }
                catch (NullPointerException e){
                }
            }
        });
    }
    @Override
    public void onBackPressed() {
        Intent a = new Intent(Intent.ACTION_MAIN);
        a.addCategory(Intent.CATEGORY_HOME);
        a.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(a);
    }
}

```

```

public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        .....
    }
}

```

```

public class ThirdActivity extends AppCompatActivity {
    private static final long COUNTDOWN_IN_MILLIS = 11000;
    private QuestionLibrary mQuestionLibrary = new QuestionLibrary();
    .....
    private CountdownTimer countDownTimer;
    private long timeleftInMillis;
    private String answer;
    private int score = 0;
    private int questionnumber =0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ....
        editText.setText(""); // initiating the play
        updatequestion();
        timeleftInMillis = COUNTDOWN_IN_MILLIS;
        startcountDown();
        b0.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View view) {
            editText.setText(editText.getText().insert(editText.getText().length(), "0"));
        }
    });
    .....
    bclr.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(editText.getText().toString().matches("")){
                Toast.makeText(getApplicationContext(), "Nothing to
clear", Toast.LENGTH_SHORT).show();
            }
            else {
                editText.setText(editText.getText().delete(editText.getText().length() - 1,
editText.getText().length()));

            }
        }
    });

    buttonenter.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            int n = mQuestionLibrary.getQuestioncount();
            if (questionnumber < n) {
                if (editText.getText().toString().equals(answer)) {
                    score = score + 10;
                    updateScore(score);
                    editText.setText("");
                    updatequestion();
                    countdownTimer.cancel();
                    timeleftInMillis = COUNTDOWN_IN_MILLIS;
                    startcountDown();
                } else if (editText.getText().toString().matches("")) {
                    Toast.makeText(getApplicationContext(), "Enter answer first",
Toast.LENGTH_SHORT).show();
                } else {
                    score = score - 5;
                    updateScore(score);
                    editText.setText("");
                    if(score <= 0){
                        textViewquestion.setText("You lost, X_Math Over!! :( \n Try
again!!\nYour score is: "+score);
                        textViewscore.setText("");
                        buttonenter.setEnabled(false);
                        b0.setEnabled(false);
                        .....
                        bclr.setEnabled(false);
                        timeleftInMillis=0;
                        updateCountDownText();
                    }
                }
            }
        }
    });

```

```

        countdownTimer.cancel();
    }
    else{
        updatequestion();
        countdownTimer.cancel();
        timeleftInMillis = COUNTDOWN_IN_MILLIS;
        startcountDown();
    }

}

} else {
    if (editText.getText().toString().equals(answer)) {
        score = score + 10;
        updateScore(score);
        editText.setText("");
        textViewquestion.setText("");
    } else if (editText.getText().toString().matches("")) {
        Toast.makeText(getApplicationContext(), "Enter answer first",
Toast.LENGTH_SHORT).show();
    } else {
        score = score - 5;
        updateScore(score);
        if(score <= 0){
            textViewquestion.setText("You lost, X_Math Over!! :( \n Try
again!!\nYour score is: "+score);
            editText.setFocusable(false);
            textViewscore.setText("");
            buttonenter.setEnabled(false);
            b0.setEnabled(false);

            .....
            bclr.setEnabled(false);
            timeleftInMillis=0;
            updateCountDownText();
            countdownTimer.cancel();

        }
        editText.setText("");
        textViewquestion.setText("");
    }
    Toast.makeText(getApplicationContext(), "End of questions",
Toast.LENGTH_SHORT).show();
    textViewquestion.setText("Congratulations!! :) \nYou have finished X_Math
with\nScore: "+score);
    editText.setFocusable(false);
    textViewscore.setText("");
    buttonenter.setEnabled(false);
    b0.setEnabled(false);

    .....
    bclr.setEnabled(false);

```

```

        timeleftInMillis=0;
        updateCountDownText();
        countDownTimer.cancel();
    }
}
});
Buttonend.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(getApplicationContext(),"U have ended X_Math\nYour score is:
"+score,Toast.LENGTH_SHORT).show();
        Intent gotoFirst = new Intent(ThirdActivity.this,FirstActivity.class);
        startActivity(gotoFirst);
    }
});
}

```

```

public class QuestionLibrary {

    //Easy Range[5,10]
    final int r1 = new Random().nextInt(3) + 2;
    final int r2 = new Random().nextInt(3) + 5;
    .....
    private String mQuestions [] = {
        //Easy questions
        r1 + " * " + r1 +" =\n",
        r2 + " * " + r2 +" =\n",
        .....
    };
    //Easy question answers
    int a1 = r1 * r1;
    int a2 = r2 * r2;
    .....
    private String mCorrectAnsweres[] = {a1+"",a2+"",.....,a20+""};
    public int getQuestioncount(){
        return mQuestions.length;
    }
    public String getQuestion(int a){
        String question = mQuestions[a];
        return question;
    }
    public String getCorrectAnswer(int a){
        String answer = mCorrectAnsweres[a];
        return answer;
    }
}

```