Report on Interface Programming subject

WRITE A MULTIPLE-CHOICE EXAMINATION PROGRAM

TEACHER: NGUYEN THI MAI TRANG

Nguyen Sinh Hung | Interface programming | April 10, 2023

Introduce

Multiple choice is a method of assessing a person's ability or knowledge through answering available questions. In this report, we will learn about how to write multiple-choice test programs through programming language
C# Windows Form .NET program

Program description

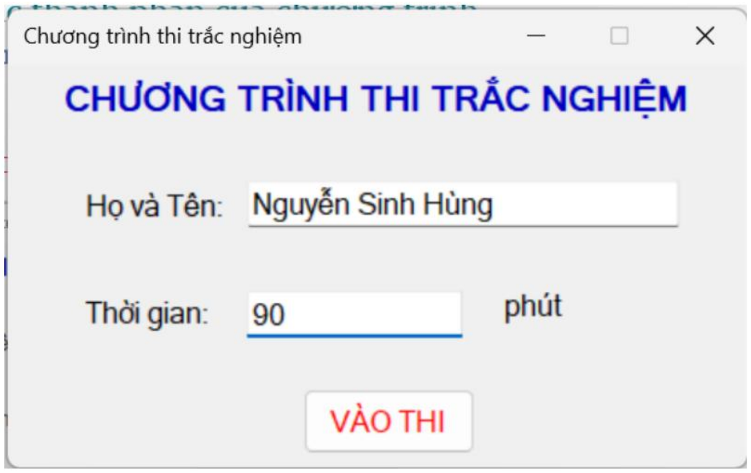
- Users will create a set of questions using a text file (.txt) according to the format program required
- Users will save the set of questions into the program and the program allows users to participate in multiple choice exams by answering questions in the selected question set.

- After the user finishes, the program displays the results (number of correct answers, number of incorrect answers, score) of the multiple choice test.

Program components

1. User interface

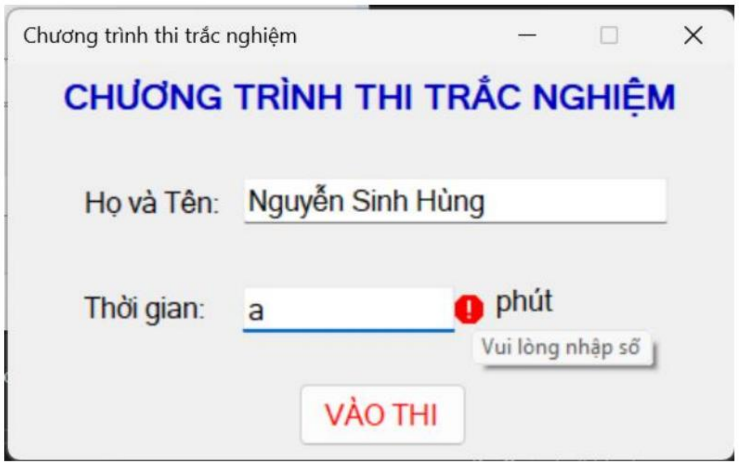
When the program is run, a table will appear where the candidate will enter their name and exam time (the program will warn if the candidate enters incorrectly). Then, click on the ENTER TEST button.


Chương trình thi trắc nghiệm

CHƯƠNG TRÌNH THI TRẮC NGHIỆM

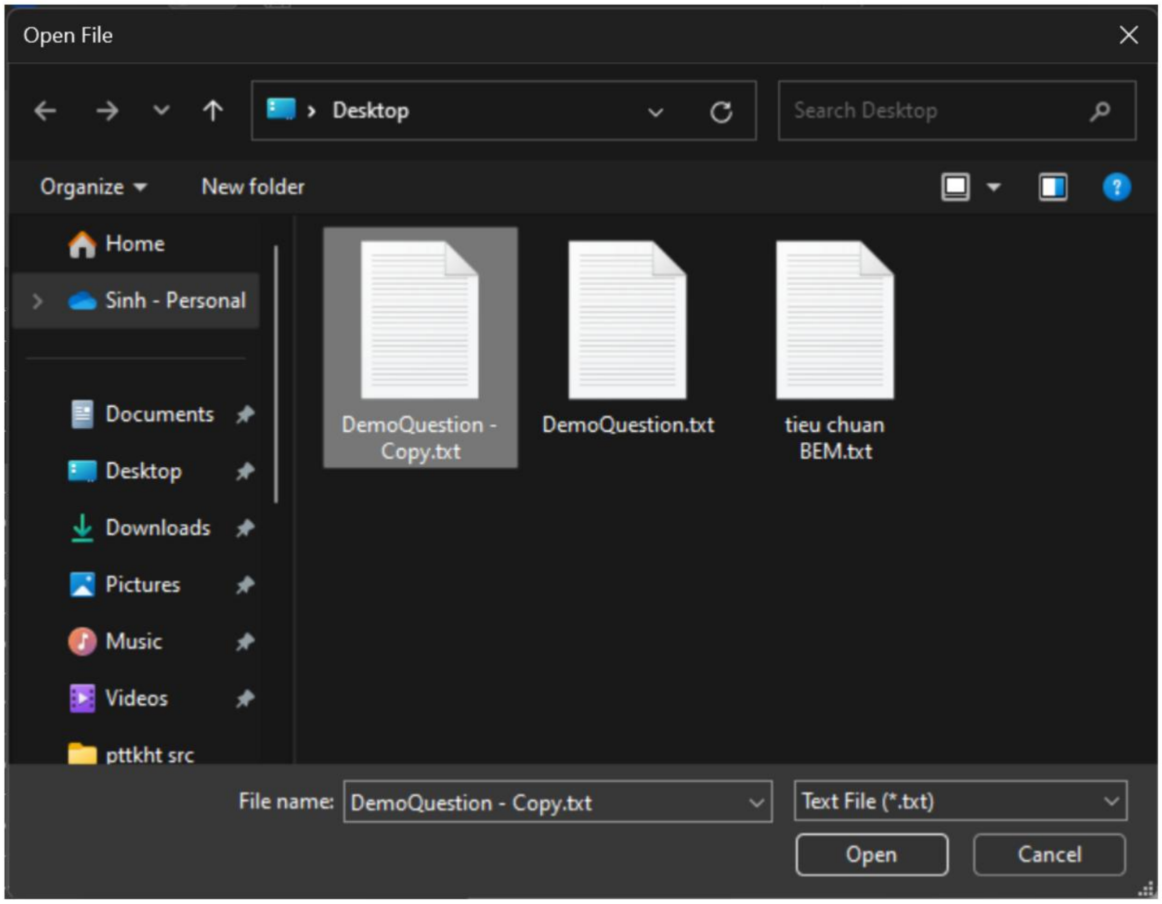
Họ và Tên: Nguyễn Sinh Hùng

Thời gian: 90 phút

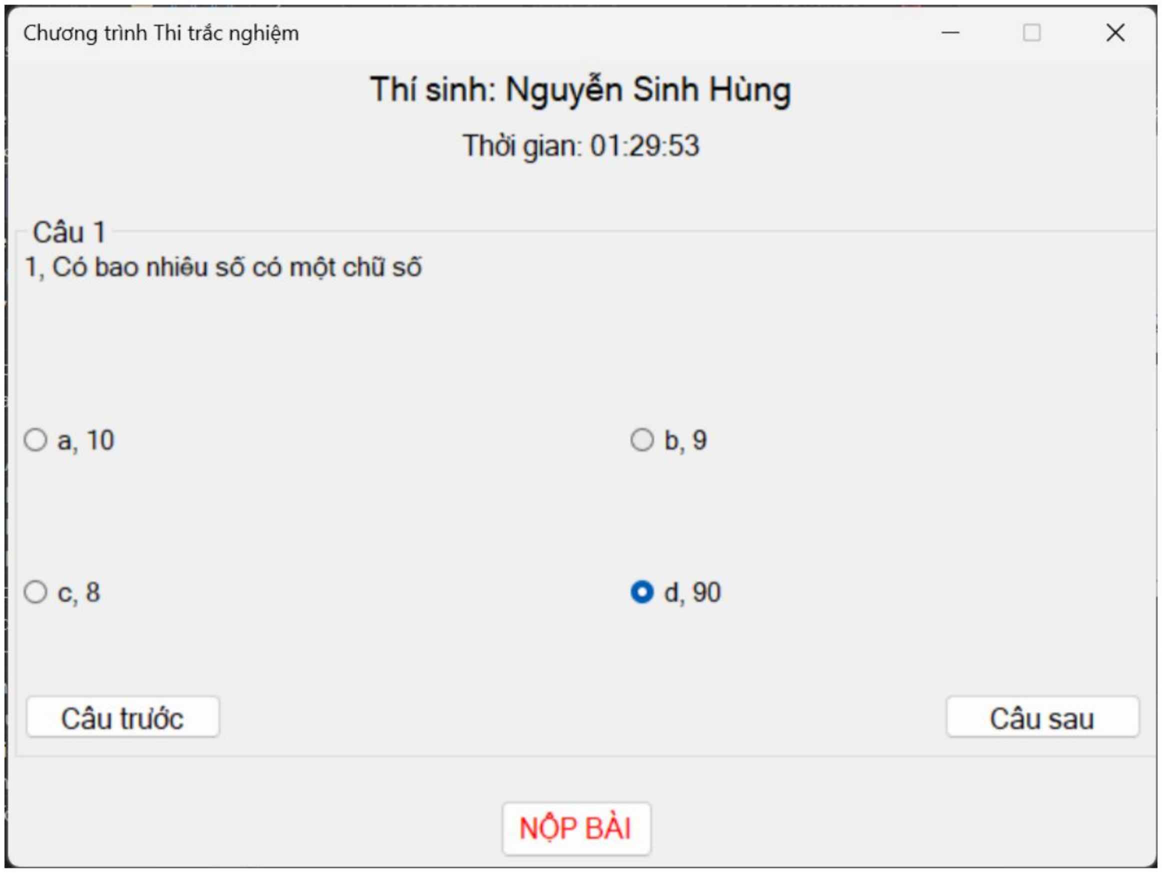


The program warns if the candidate enters the wrong number identifier

After pressing the ENTER TEST button, the program displays a window for the user to input the text file of the test into the program.



The user then selects Open, the program displays a window for candidates to begin taking the multiple-choice test.


Chương trình Thi trắc nghiệm

Thí sinh: Nguyễn Sinh Hùng

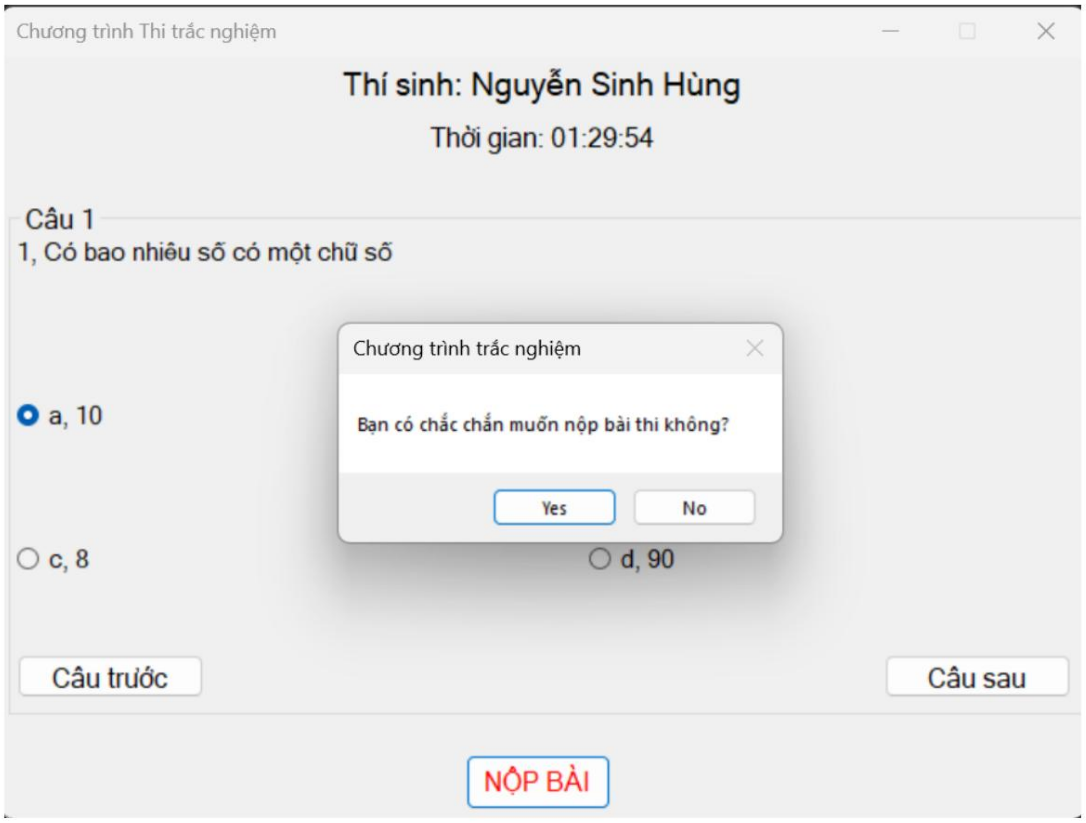
Thời gian: 01:29:53

Câu 1

1, Có bao nhiêu số có một chữ số

○ a, 10
○ b, 9
○ c, 8
● d, 90

After the candidate finishes completing the test, click on the SUBMIT button, the program displays a MessageBox


Chương trình Thi trắc nghiệm

Thí sinh: Nguyễn Sinh Hùng

Thời gian: 01:29:54

Câu 1

1, Có bao nhiêu số có một chữ số

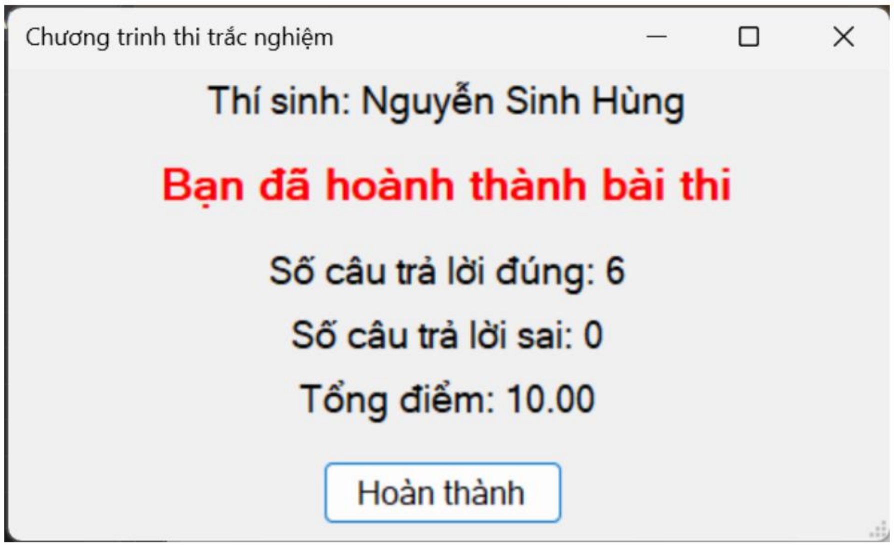
● a, 10
○ c, 8
○ d, 90

Chương trình trắc nghiệm

Bạn có chắc chắn muốn nộp bài thi không?

If you choose Yes, the program turns off the test window and displays the test results

If you do not choose No, the candidate continues to take the test.


Chương trinh thi trắc nghiệm

Thí sinh: Nguyễn Sinh Hùng

Bạn đã hoành thành bài thi

Số câu trả lời đúng: 6

Số câu trả lời sai: 0

Tổng điểm: 10.00

At the test results, the program displays the number of correct sentences, the number of incorrect sentences and calculates the candidate's score on a 10-point scale.

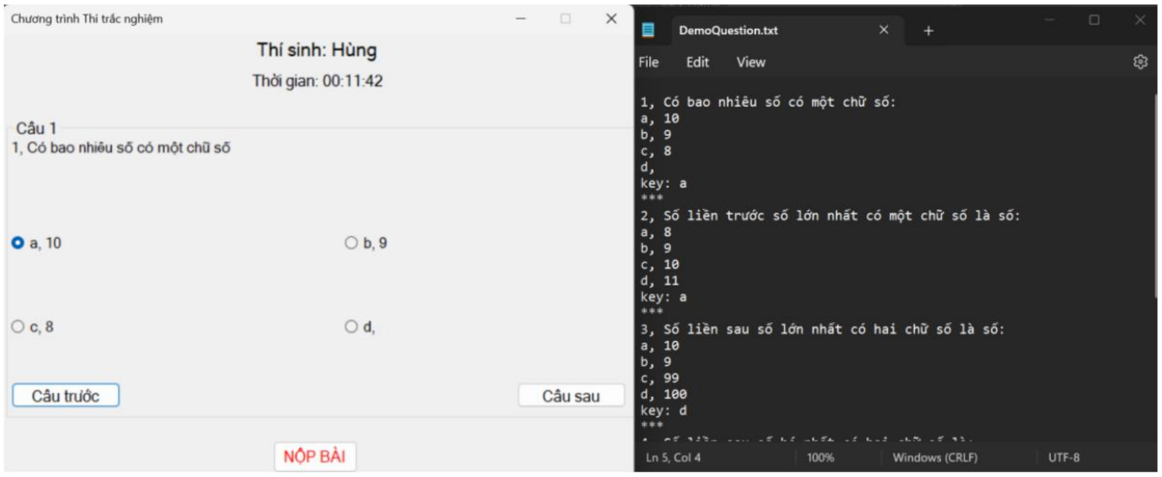
2. Processing methods

Program requirements

The program requires data to be entered in the correct file format to be readable

A multiple choice question must have a question and the answers are a,b,c,d. If the question has a missing answer (a,b,c does not have d), then you must also write d, and leave it blank

For example: sentence 1 is missing sentence d


Chương trình Thi trắc nghiệm

Thí sinh: Hùng

Thời gian: 00:11:42

Câu 1
1, Có bao nhiêu số có một chữ số

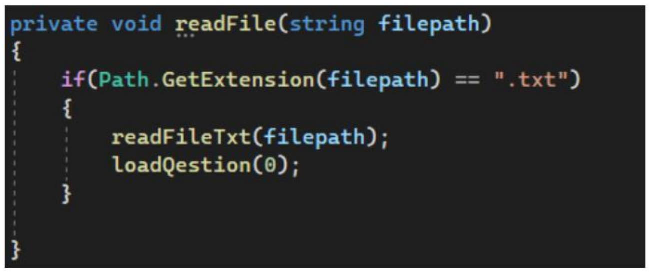
[✓ a, 10] b, 9
c, 8 d,

```
1, Có bao nhiêu số có một chữ số:
a, 10
b, 9
c, 8
d,
key: a
***
2, Số liền trước số lớn nhất có một chữ số là số:
a, 8
b, 9
c, 10
d, 11
key: a
***
3, Số liền sau số lớn nhất có hai chữ số là số:
a, 10
b, 9
c, 99
d, 100
key: d
***
```

And in the middle of each sentence there must be 3 asterisks (***) and there must be the answer to the question (key).

The method of processing input data is a text file (.txt).

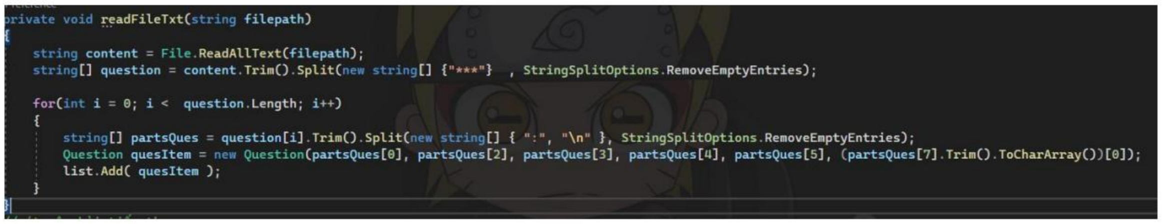
The program checks the file ending in .txt, then starts reading the text file and displaying it first question. The code design is convenient for expansion if reading additional excel files (.xlsx), word (.docx), etc.



```
private void readFile(string filepath)
{
    if(Path.GetExtension(filepath) == ".txt")
    {
        readFileTxt(filepath);
        loadQestion(0);
    }

}
```

Perform file reading



```
private void readFileTxt(string filepath)
{
    string content = File.ReadAllText(filepath);
    string[] question = content.Trim().Split(new string[] {"***"}  , StringSplitOptions.RemoveEmptyEntries);

    for(int i = 0; i <  question.Length; i++)
    {
        string[] partsQues = question[i].Trim().Split(new string[] { ":", "\n" }, StringSplitOptions.RemoveEmptyEntries);
        Question quesItem = new Question(partsQues[0], partsQues[2], partsQues[3], partsQues[4], partsQues[5], (partsQues[7].Trim().ToCharArray())[0]);
        list.Add( quesItem );
    }
}
```

The program reads file.txt all lines (ReadAllText method) and uses them Use split("***") method to cut between questions and save into questions array. Then browse through the elements of the array and continue cutting based on the characters ":" and "\n" (new line

The program has designed a Question object class that includes attributes (id, question, answer A, answer B, answer C, answer D and key). The program initializes *List<Question> list = new List<Question>()* to easily manage question objects.

Countdown time

After receiving the candidate's test time (minutes), the program will convert to how many hours, how many minutes.

For example: 90 minutes = 1 hour 30 minutes

The program uses a timer to count down the test taker's time.

After the test time expires, the program automatically closes the candidate's test window and displays the candidate's test results.

Button to perform Next Sentence and Previous Sentence

The program initializes the variable as dem = 0

When the user selects the following sentence , check if dem is less than the number of sentences. If there is a variable dem, it is increased by 1 (dem++).

When the user selects the previous sentence , check if dem is greater than 0 (dem > 0). If there is a variable dem, it is reduced by 1 (dem--).

Then load the questions of the variable.

Keep track of candidates' answers and calculate results

The program creates an empty integer array called arrAns when the form runs. After the Text file is read, we already have the number of questions in the list (in the data processing part), then initialize the array with the size of the list and set all elements of the array to -1.

Based on the variable dem, it will know which question the user is in and store the user's answer at position dem-- of the arrAns array. If the questions the user does not select, the value will still be -1.

Conversion: A is 1 ; B is 2 ; C is 3 ; D is 4

Use the CheckedChanged event for 4 radioButtons to check if radioBtn.Name is radio A then arrAns[dem] = 1, if radioBtn.Name is radio B then arrAns[dem] = 2. Similar to C and D. With the sentences If not selected, the value of the element at position dem does not change (-1).

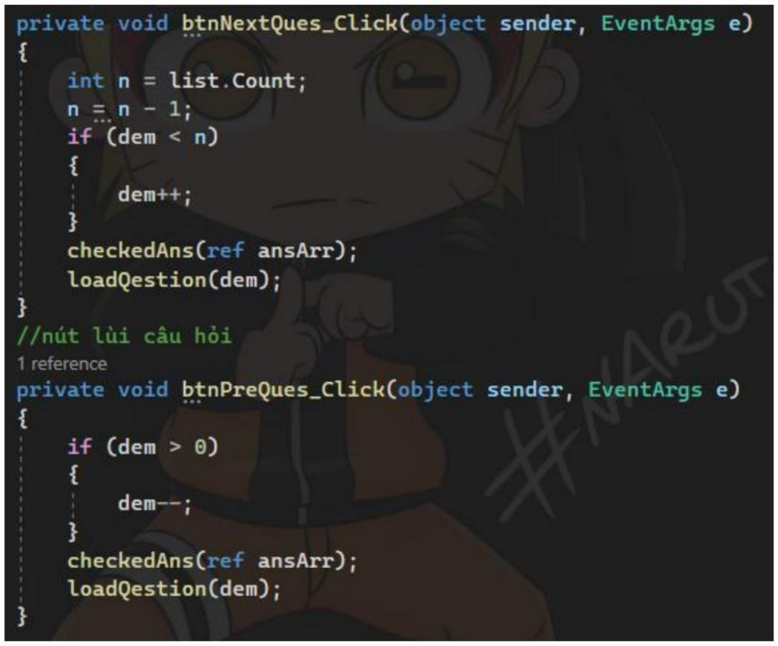
For example:

In question 1, the user chooses question B, then arrAns[0] = 2

If the user skips sentence 2, arrAns[1] does not change, so the value is still -1

Question 3: If the user chooses D then arrAns[2] = 4

If the user returns to the previous or next question but the chapter still retains the user's old answer, perform the check immediately before loading the questions of the Previous Question and Next Question buttons.



```
private void btnNextQues_Click(object sender, EventArgs e)
{
    int n = list.Count;
    n = n - 1;
    if (dem < n)
    {
        dem++;
    }
    checkedAns(ref ansArr);
    loadQestion(dem);
}
//nút lùi câu hỏi
1 reference
private void btnPreQues_Click(object sender, EventArgs e)
{
    if (dem > 0)
    {
        dem--;
    }
    checkedAns(ref ansArr);
    loadQestion(dem);
}
```

To check, we pass the reference variable of the array ansArr to the function and use a loop to go through the questions. If the element position is equal to the variable dem (number sentence the user wants to go to) then check the element at that position is 1 then radioA.Checked = true , similar to answers B, C and D.

After candidates are sure they want to submit the exam or the exam time is up. The program closes the test window. Then, we will create a variable to store the number of correct sentences (soCauDung) and initialize the keyArr array with the size of the list . The program will browse the elements of the list through the Question keys (managed by the list) to create an array of numerical answers and compare with the array of arrAns (array of candidates' answers) if the keyArr element has If the value is equal to the ansArr element, the number of correct sentences increases by 1. From there, the candidate's score is calculated.

RUN OUT OF