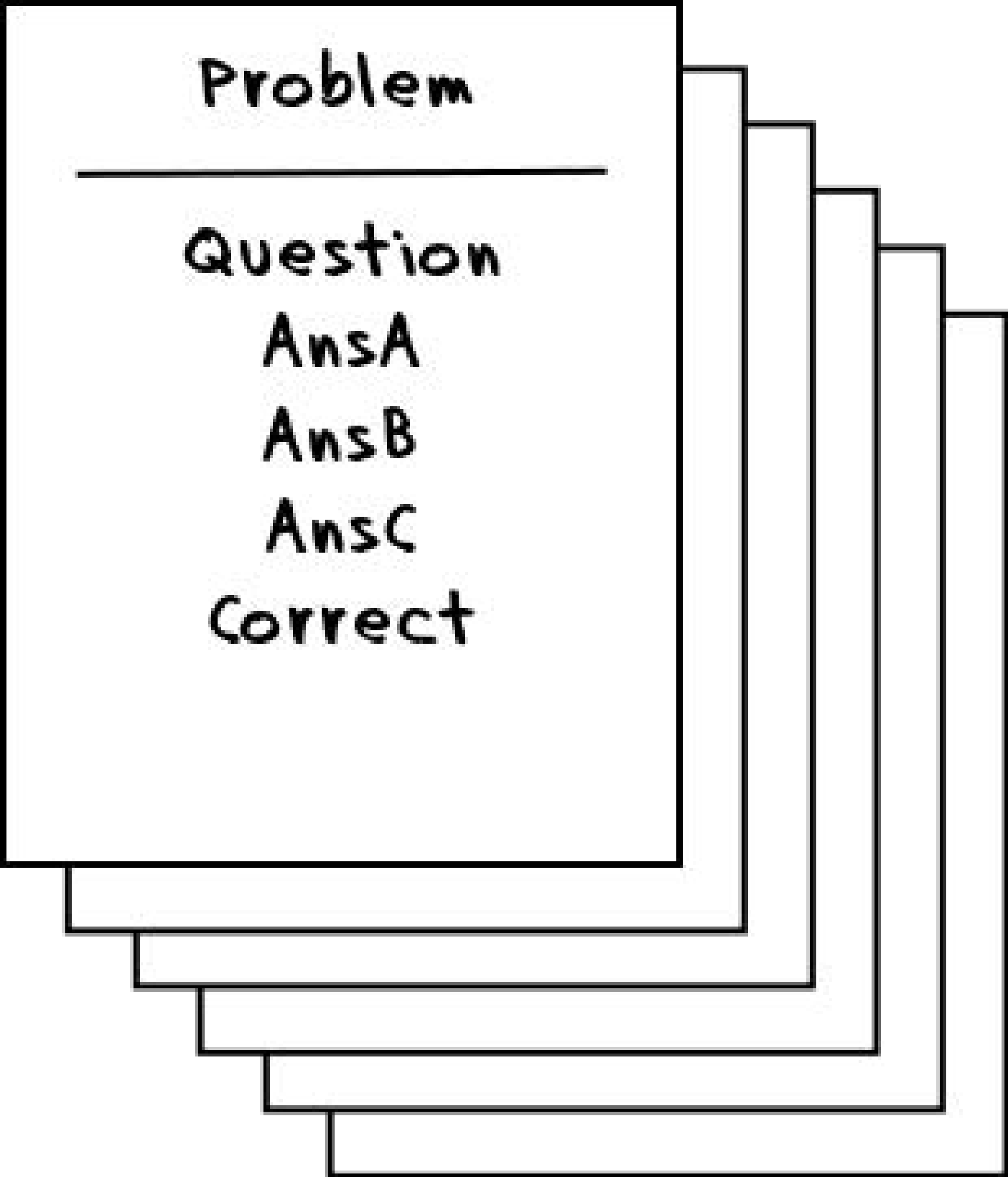
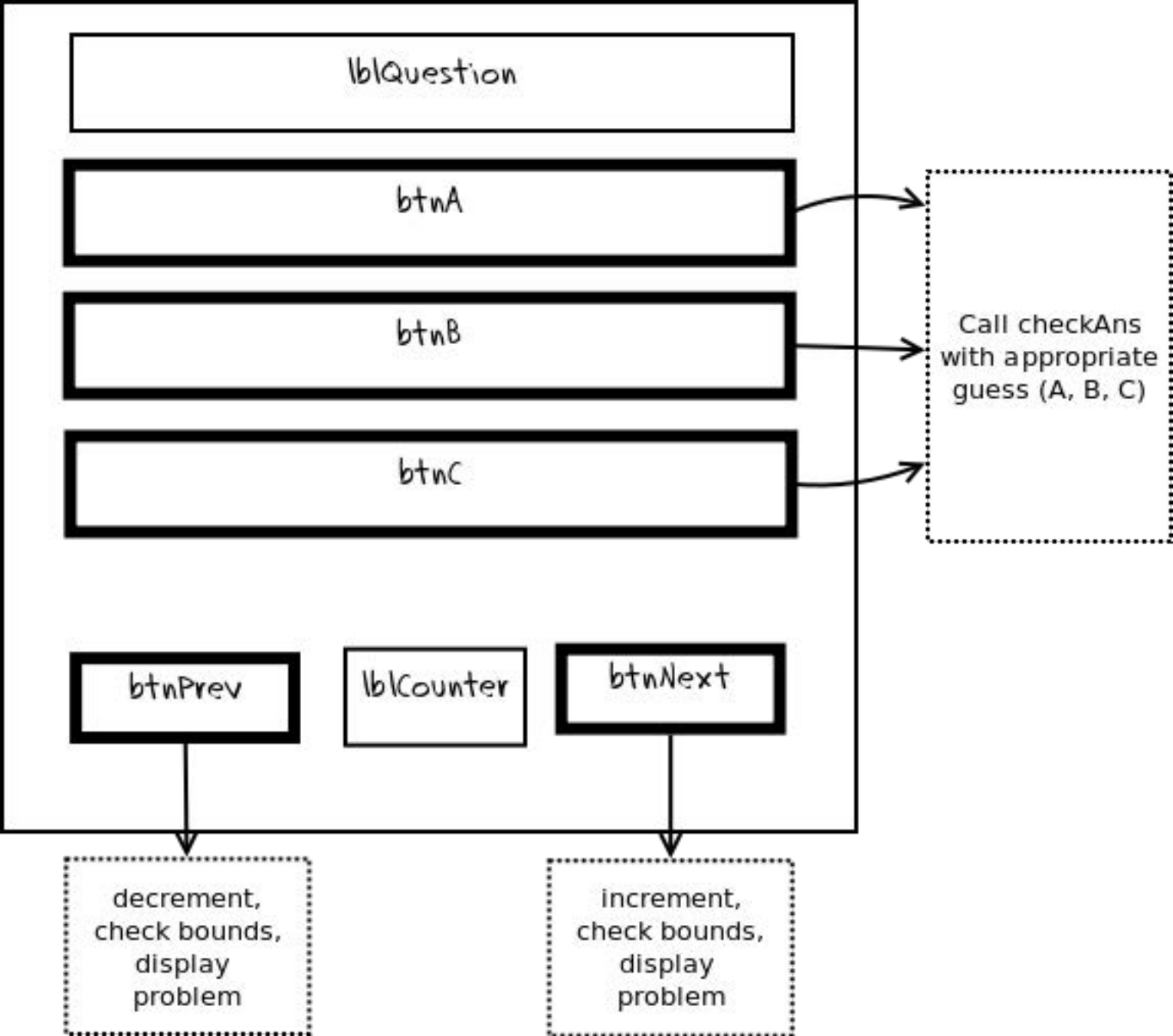

Problem
Question
AnsA
AnsB
AnsC
Correct


lblQuestion
btnA
btnB
btnC
Call checkAns with appropriate guess (A, B, C)
btnPrev
lblCounter
btnNext
decrement, check bounds, display problem
increment, check bounds, display problem

```
""" quiz.py
    example of a quiz game
    using objects for data
"""

from Tkinter import *
from tkMessageBox import *

class Problem(object):
  def __init__(self, question = "", a = "", b = "", c = "", correct = ""):
    object.__init__(self)
    self.question = question
    self.a = a
    self.b = b
    self.c = c
    self.correct = correct

class App(Tk):
  def __init__(self):
    Tk.__init__(self)

    self.problems = []
    self.counter = 0

    self.addComponents()
    self.loadProblems()

    self.showProblem(0)

    self.mainloop()

  def addComponents(self):
    """ add components to the GUI """

    self.title("Quiz")
    #force app to a fixed width
    self.grid()
    self.columnconfigure(0, minsize = 100)
    self.columnconfigure(1, minsize = 200)
    self.columnconfigure(2, minsize = 100)

    self.lblQuestion = Label(self, text = "Question")
    self.lblQuestion.grid(columnspan = 3, sticky = "we")

    self.btnA = Button(self, text = "A", command = self.checkA)
    self.btnA.grid(columnspan = 3, sticky = "we")

    self.btnB = Button(self, text = "B", command = self.checkB)
    self.btnB.grid(columnspan = 3, sticky = "we")

    self.btnC = Button(self, text = "C", command = self.checkC)
    self.btnC.grid(columnspan = 3, sticky = "we")
```

```
  self.btnPrev = Button(self, text = "prev", command = self.prev)
  self.btnPrev.grid(row = 4, column = 0)

  self.lblCounter = Label(self, text = "0")
  self.lblCounter.grid(row = 4, column = 1)

  self.btnNext = Button(self, text = "next", command = self.next)
  self.btnNext.grid(row = 4, column = 2)

def checkA(self):
  self.check("A")

def checkB(self):
  self.check("B")

def checkC(self):
  self.check("C")

def check(self, guess):
  #compares the guess to the correct answer
  correct = self.problems[self.counter].correct
  if guess == correct:
    showinfo("Quiz", "Who are you, so wise in the ways of science?")
  else:
    showinfo("Quiz", "Nii!")

def prev(self):
  self.counter -= 1
  if self.counter < 0:
    self.counter = 0
  self.showProblem(self.counter)

def next(self):
  self.counter += 1
  if self.counter >= len(self.problems):
    self.counter = len(self.problems) - 1
  self.showProblem(self.counter)

def showProblem(self, counter):
  self.lblQuestion["text"] = self.problems[counter].question
  self.btnA["text"] = self.problems[counter].a
  self.btnB["text"] = self.problems[counter].b
  self.btnC["text"] = self.problems[counter].c
  self.lblCounter["text"] = self.counter

def loadProblems(self):
  self.problems.append(Problem(
    "What is your name?",
    "Arthur, King of the Britons",
    "Roger the Shrubber",
    "Brave Sir Robin",
    "A"))

  self.problems.append(Problem(
```

```
      "What is your quest?",
      "I'm looking for the perfect pizza topping",
      "I need a shrubbery",
      "I seek the holy grail",
      "B"))

    self.problems.append(Problem(
      "We are the knights who say...",
      "nothing",
      "Your mother was a hamster",
      "NII!",
      "C"))

    self.problems.append(Problem(
      "What is your favorite color?",
      "Red",
      "Yellow",
      "Blue... No, Green!",
      "C"))

    self.problems.append(Problem(
      "What is the airspeed velocity of an unladen swallow?",
      "I don't know that - Aaaaagh!",
      "5.8 meters per second",
      "2 mph",
      "A"))


def main():
  a = App()

if __name__ == "__main__":
  main()
```