Question

You are given a C++ program (**FinalQ1.cpp**) with errors (syntax errors and/ or logical errors, if any). The program has three (3) user-defined functions as listed below:

Function Name	Description
getMenu	The function asks the users to enter a menu name and then return the menu name as a string pointer-type variable. The returned menu name is later assigned as an item of string pointer type array variable named **menus**.
getPricePcs	The function accepts two arguments. The first argument is the caption text to guide users to enter the price or the pieces' number of chicken on the menu. The second argument is either the price or the piece of chickens which were represented by an array of integer pointer type variables: **price** and **pcs**. The **getPricePcs** function assigns data entered by the users to the corresponding item of these array variables.
cheapestMenu	The function accepts three arguments that are the **menus**, **price**, and **pcs** parallel array pointer-type variables. It calculates the price for one piece of chicken on each menu (**price** / **pcs**) and then returns the index number of array items which representing the cheapest menu (menu with lowest price for one piece of chicken).

The **main** function of the program has a series of calls to **getMenu** and **getPricePcs** functions inside a loop control structure. The **menus**, **price**, and **pcs** are parallel pointer type arrays with references to their item's index was made based on variable used to control the loop (loop which used to make a series of call to **getMenu** and **getPricePcs** functions). Some of the output was produced by a call made to **cheapestMenu** function. The last part of the output produced after the index number of parallel array items representing the cheapest menu was returned by the **cheapestMenu** function. You are required to debug the errors, compile, and run the program. You are **NOT ALLOWED** to **remove** any statements in the program. You are only allowed to **update** the statements provided in the program and add a new statement(s) if absolutely necessary.

The program should produce the outputs as in **Figure 1**. ***Note:*** The values in **bold** are input by the user.

```
1  //FinalQ1.cpp
2  #include <iostream>
3  #include <string>
4  #define SIZE 3
5
6  using namespace std;
7
```

```
8   string getMenu() {
9       string *p_data = new string;
10      cout << "Menu name: ";
11      getline(cin, *p_data);
12      return p_data;
13  }
14
15  void getPricePcs(string caption, int *p_data[]) {
16      cout << caption;
17      cin >> p_data;
18  }
19
20  int cheapestMenu(string *m[], int *pr, int *pc) {
21      float pcs_price, pcs_price_lowest = 0;
22      int idx;
23
24      cout << "Check chicken price per-pcs: \n";
25
26      for (int i = 0; i < SIZE; i++) {
27          pcs_price = (float) *pr[i] / *pc[i];
28          cout << *m[i] << " - " << pcs_price << "\n";
29
30          if (pcs_price_lowest < pcs_price) {
31              pcs_price_lowest = pcs_price;
32              idx = i;
33          }
34      }
35      cout << "\n";
36
37      return idx;
38  }
39
40  // Start main function
41  int main() {
42      string *menus [SIZE];
43      int *price;
44      int *pcs [SIZE]; // pieces of chicken
45
46      // Examples of menus, price and pcs of chicken
47      // Chicken Deluxe - RM 30 - 5 pcs
48      // Happy Combo - RM 51 - 8 pcs
49      // Family Bucket - RM 75 - 15 pcs
50      for (int i = 0; i < SIZE; i++) {
51          menus[i] = getMenu();
52
53          price[i] = new string;
54          getPricePcs("Price (RM): ", price[i]);
55
56          getPricePcs("Chicken (pcs): ", pcs[i]);
57
58          cin.ignore(); //  need this as we mix the use of getline and cin
59
60          cout << "\n";
61      }
62
63      int idx_cheapest = cheapestMenu(menus, price, pcs);
64
65      cout << "Cheapest menu is " << *menus[idx_cheapest]
66           << " priced at RM " << *price[idx_cheapest] << " for "
```

```
67          << pcs[idx_cheapest] << " pieces of chicken\n\n";
68
69      // delete array data from memory
70      for (int i = 0; i < SIZE; i++) {
71          delete menus[i];
72          delete price[i];
73          delete pcs[i];
74      }
75
76      return 0;
77  }
```

```
Menu name: Chicken Deluxe
Price (RM): 30
Chicken (pcs): 5

Menu name: Happy Combo
Price (RM): 51
Chicken (pcs): 8

Menu name: Family Bucket
Price (RM): 75
Chicken (pcs): 15

Check chicken price per-pcs:
Chicken Deluxe - 6
Happy Combo - 6.375
Family Bucket - 5

Cheapest menu is Family Bucket priced at RM 75 for 15 pieces of chicken
```

Figure 1: The example of output