

## BITP 1113: PROGRAMMING TECHNIQUES

### 1BITS

### Lab 9

### Array Part 2

#### Exercise 1

Write a program to let a user chooses the type of report, as stated in Table 1. These reports are generated based on the number of passengers data, as shown in Table 2.

**Table 1: Number of Passenger Reports**

Report No	Report Type
1	View number of passengers according to bus and day selected
2	View number of passengers according to bus only
3	View all number of passengers available data

**Table 2: Number of Passenger According to Bus and Day**

Bus Plate No	Mon	Tue	Wed	Thu	Fri	Sat	Sun
MAW 4572	40	35	37	32	44	44	44
WJV 441	44	35	30	32	44	37	35
MBV 6560	40	40	40	40	40	40	40

1. Analyse input, process and output.

<b>View Report No 1</b>	<b>Input</b> : plateNo and day <b>Process</b> : - <b>Output</b> : number of passengers based on plateNo and day (individual elements in 2-D passenger array)
<b>View Report No 2</b>	<b>Input</b> : plateNo <b>Process</b> : - <b>Output</b> : day and number of passengers based on plateNo (elements in the a row of 2-D passenger array)
<b>View Report No 3</b>	<b>Input</b> : - <b>Process</b> : - <b>Output</b> : plateNo, day and all number of passengers (elements in 2-D passenger array)

2. Create a new project named `PassingArray`
3. At **Solution Explorer**, right-click **Source Files** folder and add new item named `Passenger.cpp`.
4. Write the default code.

5. Declare array size declarators BUS and DAY with initial values 3 and 7, respectively (Line 4). Then initialise 1-D array variables which are busList, dayList and passenger (Line 7 to 14).

```

2   using namespace std;
3
4   const int BUS = 3, DAY = 7; // Initialize array size declarators
5
6   int main() {
7       // Initialize buses plate number
8       string busList[BUS] = { "MAW 4572", "WJV 441", "MBV 6560" };
9       // Initialize days in a week
10      string dayList[DAY] = { "Monday", "Tuesday", "Wednesday",
11                              "Thursday", "Friday", "Saturday", "Sunday" };
12      // Initialize number of passengers
13      int passenger[BUS][DAY] = { {40, 35, 37, 32, 44, 44, 44},
14                                  {44, 35, 30, 32, 44, 37, 35}, {40, 40, 40, 40, 40, 40, 40} };
15      return 0;
16  }

```

6. Write a function call to display the type of reports and accept selection made by a user then return the selection for further actions in calling function (Line 15).

```

13      int passenger[BUS][DAY] = { {40, 35, 37, 32, 44, 44, 44},
14                                  {44, 35, 30, 32, 44, 37, 35}, {40, 40, 40, 40, 40, 40, 40} };
15      operation = menuOperation(); // Choose the operation for the array
16      return 0;

```

7. Declare the variable operation (Line 8) and write a function prototype for menuOperation() (Line 5).

```

4   const int BUS = 3, DAY = 7; // Initialize array size declarators
5   int menuOperation(); // Function prototype
6
7   int main() {
8       int operation; // Declare the variable
9       // Initialize buses plate number

```

8. Write menuOperation() function definition (Line 48 to 59).

```

48  // Function definition: Choose which data to display
49  int menuOperation() {
50      int choice;
51      cout << "LIST OF REPORTS\n";
52      cout << "1. View number of passengers according to bus and day selected\n";
53      cout << "2. View number of passenger according to bus only\n";
54      cout << "3. View all data\n";
55      cout << "Press any other key to exit the program\n";
56      cout << "Your choice: ";
57      cin >> choice;
58      return choice;
59  }

```

9. At `main()` function, if the return value, i.e. `operation` is 1, it indicates the user wants to view a number of passengers according to specific bus and day (Line 19). So, we need to write code to accept input of bus plate number and day selected. Then, we use these inputs to display the number of passengers. Therefore, we have to create three function calls which are:
- `menuBus()`: pass/receive the whole 1-D array of bus plate numbers to display it in the called function, accept bus plate number selected by a user and return it to the calling function (Line 20 and 21).
  - `menuDay()`: pass/receive the whole 1-D array of days in a week to display it in the called function, accept day selected by a user and return it to the calling function (Line 22 and 23).
  - `viewBusDay()`: pass/receive the individual element of 2-D array which represent the number of passengers for a particular bus and day and display this element in the called function (Line 24 and 25).

```
18 operation = menuOperation(); // Choose the operation for the array
19 if (operation == 1) {
20     // function call: passing the whole 1-D array
21     plateNo = menuBus(busList); // get the bus plate number
22     // function call: passing the whole 1-D array
23     day = menuDay(dayList); // get the day
24     // function call: passing individual element of 2-D array
25     viewBusDay(passenger[plateNo][day]);
26 }
27 return 0;
```

10. Declare variables `plateNo` and `day` (Line 12) and write function prototypes for `menuBus()`, `menuDay()` and `viewBusDay()` (Line 7 to 9).

```
6 int menuOperation(); // Function prototype
7 int menuBus(string b[]); // Function prototype
8 int menuDay(string d[]); // Function prototype
9 void viewBusDay(int p); // Function prototype
10
11 int main() {
12     int operation, plateNo, day; // Declare the variables
13     // Initialize buses plate number
```

11. Write menuBus () function definition (Line 64 to 80).

```
64 // Function definition: to display list of bus plate numbers
65 // and let the user choose the bus plate number
66 int menuBus(string b[]) {
67     int choice;
68     do{
69         cout << "\nLIST OF BUSES\n";
70         // Display plate numbers. Output element-by-element
71         for (int i = 0; i < BUS; i++) {
72             cout << i + 1 << ". " << b[i] << endl;
73         }
74         cout << "Choose the bus: ";
75         cin >> choice;
76         if (choice < 1 || choice > 3)
77             cout << "Invalid choice. Please try again.";
78     } while (choice < 1 || choice > 3); // loop if invalid input
79     return choice - 1; // return the subscript of element
80 }
```

12. Write menuDay () function definition (Line 82 to 98).

```
82 // Function definition: to display list of days in a week
83 // and let the user choose the day
84 int menuDay(string d[]) {
85     int choice;
86     do {
87         cout << "\nDAYS IN A WEEK\n";
88         // Display days in a week. Output element-by-element
89         for (int i = 0; i < DAY; i++) {
90             cout << i + 1 << ". " << d[i] << endl;
91         }
92         cout << "Choose the day: ";
93         cin >> choice;
94         if (choice < 1 || choice > 7)
95             cout << "Invalid choice. Please try again.";
96     } while (choice < 1 || choice > 7); // loop if invalid input
97     return choice - 1; // return the subscript of element
98 }
```

13. Write viewBusDay () function definition (Line 81 to 85).

```
81 // Function definition: display number of passengers
82 // for a particular bus and day.
83 void viewBusDay(int p) {
84     cout << "The number of passenger is " << p << " passengers.\n\n";
85 }
```

14. At main () function, if the return value, i.e. operation is 2, it indicates the user wants to view the number of passengers for a specific bus in a week (Line 19). So, we need to write code to accept input of bus plate number. Then, we use this plate number to display the number of passengers for a week. Therefore, we have to create two function calls which are:

- a. `menuBus()` : pass/receive the whole 1-D array of bus plate numbers to display it in the called function, accept bus plate number selected by a user and return it to the calling function (Line 33).
- b. `viewBus()` : pass/receive a row of 2-D passenger array and the whole 1-D `dayList` array. The passenger will be used to display the number of passengers for the selected bus in a week. The row represents the bus plate number. The `dayList` will be used to display the days in a week while printing the number of passengers (Line 34 and 35).

```

30      viewBusDay(passenger[plateNo][day]);
31    }
32    else if (operation == 2) {
33      plateNo = menuBus(busList); // choose the bus
34      // function call: passing a row (2-D), whole 1-D array
35      viewBus(passenger[plateNo], dayList);
36    }
37    return 0;

```

15. Reuse the existing `menuBus()` function prototype and function definition. Write function prototypes for `viewBus()` (Line 10).

```

9      void viewBusDay(int p); // Function prototype
10     void viewBus(int vB[], string vDay[]); // Function prototype
11
12     int main() {

```

16. Write `viewBus()` function definition (Line 93 to 100):

- a. `vB[]` will point to the passing `passenger[plateNo]`. `plateNo` represents a row.
- b. `vDay[]` will point to the `dayList` array.
- c. `vDay[i]` is the element of `dayList` array, which is the day such as "Monday".
- d. `vB[i]` is the element of `passenger[plateNo]` which is a number of passengers such as "40".

```

93     // Function definition: display number of passengers
94     // for a particular bus.
95     void viewBus(int vB[], string vDay[]) {
96         cout << "Number of Passenger in a Week.\n";
97         for (int i = 0; i < DAY; i++)
98             cout << vDay[i] << ": " << vB[i] << " passengers\n";
99         cout << endl;
100    }

```

17. At `main()` function, if the return value, i.e. operation is 3, it indicates the user wants to view all number of passengers available data (Line 36). So, we will create a function call to pass/receive the whole 2-D passenger array then displaying each element of the array (Line 37 and 38).

```
34     viewBus(passenger[plateNo], dayList);
35 }
36 else if (operation == 3) {
37     // function call: passing the whole array
38     viewAll(passenger, busList, dayList);
39 }
40 return 0;
```

18. Write function prototypes for `viewAll()` (Line 11).

```
10 void viewBus(int vB[], string vDay[]); // Function prototype
11 void viewAll(int vA[][DAY], string bA[], string dA[]); // Function prototype
12
13 int main() {
```

19. Write `viewAll()` function definition (Line 107 to 120):

- `vA[]` will point to the passing passenger array.
- `bA[]` will point to the `busList` array.
- `dA[]` will point to the `dayList` array.
- `bA[i]` is the element of `busList` array, which is bus plate number such as “MAW 4572”.
- `dA[j]` is the element of `dayList` array, which is the day such as “Monday”.
- `vA[i][j]` is the element of passenger array, which is the number of passengers such as “40”.

```
107 // Function definition: display number of passengers
108 // for all buses in a week.
109 void viewAll(int vA[][DAY], string bA[], string dA[]) {
110     cout << "Number of Passenger: Bus X Day \n";
111     for (int i = 0; i < BUS; i++) {
112         cout << "Plate Number : " << bA[i] << endl; // display plate number
113         for (int j = 0; j < DAY; j++) {
114             // display day and number of passengers
115             cout << dA[j] << ": " << vA[i][j] << " passengers\n";
116         }
117         cout << endl;
118     }
119     cout << endl;
120 }
```

20. At `main()` function, if the return value, i.e. operation is other than 1, 2 or 3, it indicates the user wants to exit the program. So, the program will display the termination program message and stop executing (Line 41 and 42).

```
39      viewAll(passenger, busList, dayList);
40    }
41    else
42      cout << "Program terminated";
43    return 0;
44  }
```

21. We will let the user choose the operation repeatedly.

- a. Insert code to start the loop (Line 23).

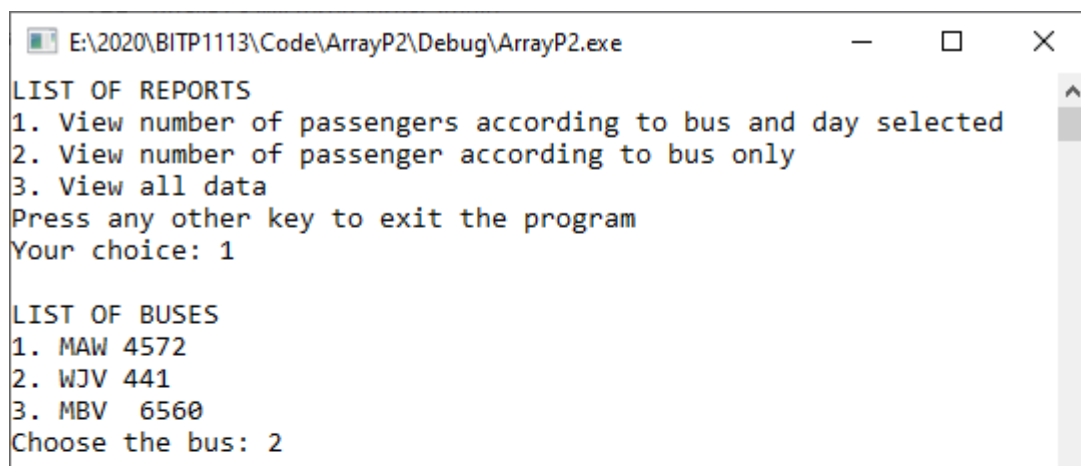
```
20  // Initialize number of passengers
21  int passenger[BUS][DAY] = { {40, 35, 37, 32, 44, 44, 44},
22    {44, 35, 30, 32, 44, 37, 35}, {40, 40, 40, 40, 40, 40, 40} };
23  do {
24    operation = menuOperation(); // Choose the operation for the array
25    if (operation == 1) {
```

- b. Insert code to end the loop (Line 44).

```
42    else
43      cout << "Program terminated";
44    } while (operation >= 1 && operation <= 3);
45    return 0;
46  }
```

22. Compile and run the program.

### Example output



```
E:\2020\BITP1113\Code\ArrayP2\Debug\ArrayP2.exe
LIST OF REPORTS
1. View number of passengers according to bus and day selected
2. View number of passenger according to bus only
3. View all data
Press any other key to exit the program
Your choice: 1

LIST OF BUSES
1. MAW 4572
2. WJV 441
3. MBV 6560
Choose the bus: 2
```

DAYS IN A WEEK

1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday

Choose the day: 5

The number of passenger is 44 passengers.

LIST OF REPORTS

1. View number of passengers according to bus and day selected
2. View number of passenger according to bus only
3. View all data

Press any other key to exit the program

Your choice: 2

LIST OF BUSES

1. MAW 4572
2. WJV 441
3. MBV 6560

Choose the bus: 3

Number of Passenger in a Week.

Monday: 40 passengers

Tuesday: 40 passengers

Wednesday: 40 passengers

Thursday: 40 passengers

Friday: 40 passengers

Saturday: 40 passengers

Sunday: 40 passengers

LIST OF REPORTS

1. View number of passengers according to bus and day selected
2. View number of passenger according to bus only
3. View all data

Press any other key to exit the program

Your choice: 3\_

Number of Passenger: Bus X Day

Plate Number : MAW 4572

Monday: 40 passengers

Tuesday: 35 passengers

Wednesday: 37 passengers

Thursday: 32 passengers

Friday: 44 passengers

Saturday: 44 passengers

Sunday: 44 passengers

Plate Number : WJV 441

Monday: 44 passengers

Tuesday: 35 passengers

Wednesday: 30 passengers

Thursday: 32 passengers

Friday: 44 passengers

Saturday: 37 passengers

Sunday: 35 passengers



```
Plate Number : MBV 6560
```

```
Monday: 40 passengers
```

```
Tuesday: 40 passengers
```

```
Wednesday: 40 passengers
```

```
Thursday: 40 passengers
```

```
Friday: 40 passengers
```

```
Saturday: 40 passengers
```

```
Sunday: 40 passengers
```

```
LIST OF REPORTS
```

```
1. View number of passengers according to bus and day selected
```

```
2. View number of passenger according to bus only
```

```
3. View all data
```

```
Press any other key to exit the program
```

```
Your choice: q
```

```
Program terminated
```

## Lab Attendance Week 11

### Question

Modify the Exercise code to:

- a) display the total number of passengers when a user chooses either operation 2 or 3.
- b) View number of passengers according to the day only.

Submit this exercise at ULearn before 12.00 p.m. 24 December 2020 (Thursday).

*~The harder you work for something, the greater you'll feel when you achieve it~*

*- Anonymous*