ASSAIGNMENT 04

Name: Hasin Araf(A20EC4024)
Ibtisam Ahmed Promit(A20EC4027)
Md Faridul Islam(A20EC4030)

Lecturer name: Dr. Goh Eg su.

PART A [70 MARKS]

QUESTION 1 [4 Marks]

Program 1 is the program that is used by the parking attendant to calculate the fee for parking based on the hours and rates. The parking building consists of employee and non-employee car with different type of rates:

(a) Employee of the company: RM 1.50 per hour

(b) Non-employee: RM 3.00 per hour

Based on the comments given in **Program 1**, fill in the blank lines with appropriate C++ statements. The program should generate the output as shown in **Figure 1**. *Note:* The **bold texts** in **Figure 1** indicate input from the user. (4 marks)

```
The number of cars: 2

Hours parked for car 1: 3
Hourly pay rate for car 1: 1.50

Hours parked for car 2: 5
Hourly pay rate for car 2: 3.00

Here is the total fee for each car:
Car [1] = RM4.50
Car [2] = RM15.00
```

Figure 1: Example runs of the Program 1

```
1  ff Program 1
2  #include<iostream>
3  #include<iomanip>
4  using namespace std;
5
```

```
6   int main()
7   {
8     const int no_cars = 2; //Constant for the array size
9
10    int hours[2];//Declare an array named hours and rate to
      allocate 2
11    double rate[2];//elements of type integer and double
      respectively.
12    //Use constant declared in line 8.
```

```
13    cout << "The number of cars: " << no_cars;
14    cout << endl;
15
16    for (int index = 0; index < no_cars; index++)
17    {
18      cout << "\nHours parked for car " << (index+1) << ": ";
19      //Read a value and assign it into array named hours
20      cin >> hours[index];
21      cout << "Hourly pay rate for car " << (index+1) << ": ";
22      //Read a value and assign it into array named rate
23      cin >> rate[index];
24    }
25    cout << "\nHere is the total fee for each car:\n";
26    cout << fixed << showpoint << setprecision(2);
27
28    for (int index = 0; index < no_cars; index++)
29    {
30       //Calculate a value of totalPay (fee for parking)
31       double totalPay = hours[index]*rate[index];
32       cout << "Car [" << (index + 1);
33       cout << "] = RM" << totalPay << endl;
34    }
35     return 0;
36
37
38
}
```

QUESTION 2 **[16 Marks]**

Given the declaration and initialization of some parallel arrays as in the following code segment:

```
string patients[] = {"Wendy", "Kumar", "Ros", "Mael"};
int age[] = {25, 43, 32, 54};
double averageSugarLevel[4];
```

Based on the concept of arrays and using loops, answer the following questions:

(a) Define a two-dimensional array named **`sugarLevel`** which has large enough space to hold the data in the following table. Initialize the array with values from the table.

(2 marks)

4.5	5.7	6.8
6.2	7.1	7.3
5.1	6.3	8.0
7.5	8.5	7.3

Answer:

```
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

int main()
{
   string patients[] = {"Wendy", "Kumar", "Ros", "Mael"};
   int age[] = {25, 43, 32, 54};
   double averageSugarLevel[4];

   double sugarLevel[4][3];

   // table row 1 all values entered
   sugarLevel[0][0] = 4.5;
   sugarLevel[0][1] = 5.7;
   sugarLevel[0][2] = 6.8;

   // table row 2 all values entered
   sugarLevel[1][0] = 6.2;
   sugarLevel[1][1] = 7.1;
   sugarLevel[1][2] = 7.3;
```

```
    // table row 3 all values entered
    sugarLevel[2][0] = 5.1;
    sugarLevel[2][1] = 6.3;
    sugarLevel[2][2] = 8.0;

    // table tow 4 all values entered
    sugarLevel[3][0] = 7.5;
    sugarLevel[3][1] = 8.5;
    sugarLevel[3][2] = 8.3;
    return 0;
}
```

(b) Write **a code segment** that prints the patient names along with their ages. The output should be as follows: (3 marks)

```
Age of Patients
--------------------
Wendy      25
Kumar      43
Ros        32
Mael       54
```

Answer:

```
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

int main()
{
   string patients[] = {"Wendy", "Kumar", "Ros", "Mael"};
   int age[] = {25, 43, 32, 54};
   double averageSugarLevel[4];

   double sugarLevel[4][3];

   cout<< "Age of Patients" << endl;
   cout<< "---------------" << endl;
   for(int i=0; i<4; i++)
   {
      cout<< left << setw(10) << setfill(' ') << patients[i] << age[i] << endl;
   }

   return 0;
}
```

(c) Write **a code segment** that produces the following output from the array named **`sugarLevel`** that you defined in (a). (4 marks)

```
Name          Breakfast   Lunch   Dinner
Wendy           4.5         5.7     6.8
Kumar           6.2         7.1     7.3
Rose            5.1         6.3     8.0
Mael            7.5         8.5     7.3
```

Answer:

```
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

int main()
{
   string patients[] = {"Wendy", "Kumar", "Ros", "Mael"};
   int age[] = {25, 43, 32, 54};
   double averageSugarLevel[4];

   double sugarLevel[4][3];

   // table row 1 all values entered
   sugarLevel[0][0] = 4.5;
   sugarLevel[0][1] = 5.7;
   sugarLevel[0][2] = 6.8;

   // table row 2 all values entered
   sugarLevel[1][0] = 6.2;
   sugarLevel[1][1] = 7.1;
   sugarLevel[1][2] = 7.3;

   // table row 3 all values entered
   sugarLevel[2][0] = 5.1;
   sugarLevel[2][1] = 6.3;
   sugarLevel[2][2] = 8.0;

   // table tow 4 all values entered
   sugarLevel[3][0] = 7.5;
   sugarLevel[3][1] = 8.5;
   sugarLevel[3][2] = 8.3;

   cout<< left << setw(15) << setfill(' ') << "Name" << setw(12) << "Breakfast" << setw(8) <<
"Lunch" << "Dinner" << endl;
   for(int j=0; j<4; j++)
   {
      cout<< left << setw(15) << setfill(' ') << patients[j];
      for(int k=0; k<3; k++)
      {
         if(k==0)
         {
            cout<< setw(12) << sugarLevel[j][k];
         }
         else
         {
            cout<< setw(8) << sugarLevel[j][k];
         }
```

```
        }
        cout<<endl;

    }
    cout<< endl;
    return 0;
}
```

(d) Write a **code segment** to calculate the average sugar level of the patient. The code also store the average sugar level value in the array **`averageSugarLevel`** and print the list of the patients' average sugar level value as follows. (7 marks)

```
Sugar Level of Patients
-------------------------------
Wendy      25
Kumar      43
Ros        32
Mael       54
```

Answer:

```
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

int main()
{
   string patients[] = {"Wendy", "Kumar", "Ros", "Mael"};
   int age[] = {25, 43, 32, 54};
   double averageSugarLevel[4];

   double sugarLevel[4][3];

   // table row 1 all values entered
   sugarLevel[0][0] = 4.5;
   sugarLevel[0][1] = 5.7;
   sugarLevel[0][2] = 6.8;

   // table row 2 all values entered
   sugarLevel[1][0] = 6.2;
   sugarLevel[1][1] = 7.1;
   sugarLevel[1][2] = 7.3;

   // table row 3 all values entered
   sugarLevel[2][0] = 5.1;
   sugarLevel[2][1] = 6.3;
   sugarLevel[2][2] = 8.0;

   // table tow 4 all values entered
   sugarLevel[3][0] = 7.5;
   sugarLevel[3][1] = 8.5;
   sugarLevel[3][2] = 8.3;

   cout<<endl;


   for(int i=0; i<4; i++)
   {
      double sum = 0.0;
```

```
        for(int j=0; j<3; j++)
        {
            sum = sum + sugarLevel[i][j];
        }
        averageSugarLevel[i] = sum;
    }

    cout<< "Sugar Level of Patients" << endl;
    cout<< "-----------------------" << endl;
    for(int i=0; i<4; i++)
    {
        cout<< left << setw(10) << setfill(' ') << patients[i] << averageSugarLevel[i]
<< endl;
    }

    return 0;
}
```

QUESTION 3 **[11 Marks]**

(a) Name at least two member functions associated with **`iostream`** object. (2 marks)

Answer:

Two member functions associated with iostream object:

- get()
- put()

Uses:

```
char x;
 // used to read a single character
   cin.get(x);

   // used to dsiplay single character onto the screen.
   cout.put(x);
```

(b) Suppose two file stream objects named **`fin`** and **`fout`** have been declared in the code as follows:

```
ifstream fin;
ofstream fout;
```

Complete the code so that the program can copy 10 numbers from a file named **`file1.dat`** and save the numbers to another file named **`file2.dat`** (copy 10 numbers between files). The program should also include an error checking for opening the files. (9 marks)

Answer:

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
ifstream fin;
ofstream fout;
fin.open("file1.dat"); //file1.dat opened in input mode
if(!fin)//if file1 does not open
{
cout<<"\nfile1.dat not opened\n ";
exit(0);
}
fout.open("file2.dat"); //file2.dat opened in output mode
if(!fout)//if file2 does not open
{
cout<<"\nfile2.dat not opened\n ";
exit(0);
}
int ch; // to read numbers
fin>>ch; // read first number from file 1.dat
int COUNT=0; //to count how many numbers
while(COUNT!=10) // till count is not 10
{
fout<<ch<<" "; //number written to file2.dat
fin>>ch; //next number read from file1.dat
COUNT++; //counting increased
}
fin.close(); //connection closed
fout.close(); //connection closed
cout<<"\nNumbers Copied ";
return 0;
}
```

QUESTION 4 **[4 Marks]**

Based on the **Program 2**, write the output into **Table 1**. *Note:* one cell represents one space per digit or character. (4 marks)

```
1   ff Program 2
2   ffThis program write three rows of numbers to a file
3   #include <iostream>
4   #include <fstream>
5   #include <iomanip>
6   using namespace std;
7
8   int main ()
9   {
10      const int ROWS = 3;
11      const int COLS = 3;
12      double nums [ROWS][COLS] = {1501, 3.579, 984,
13                                  5.3, 91, 1979,
14                                  24, 606, 11.0287};
15      fstream outFile ("Jadual.txt", ios::out);
16
17      for (int row = 0; row < ROWS; row++)
18      {
19          for (int col = 0; col < COLS; col++)
```

```
20          {
21              outFile << setprecision (1) << fixed << showpoint;
22              outFile << setw(6) << nums [row][col];
23          }
24          outFile << endl;
25      }
26      outFile.close();
27      return 0;
28  }
```

Table 1: Output file named **`Jadual.txt`**

Answer:

1	5	0	1	.	0				3	.	6		9	8	4	.	0
			5	.	3			9	1	.	0	1	9	7	9	.	0
		2	4	.	0		6	0	6	.	0			1	1	.	0

QUESTION 5 **[4 Marks]**

As you are travelling to the United States, it is important to know the temperature for you to be able to prepare what to wear for the day. Apparently in the United States, they use Fahrenheit and not Celsius. A conversion program that will ask for a temperature in Fahrenheit, convert that value to Celsius, and display the result, would definitely become handy. Complete the **Program 3**. You need to write the function **`convert`**. *Note: C* = 5 / 9 × (*F* - 32). Do not modify the function **`main`**.

Answer: (4 marks)

```
1   ff Program 3
2   #include <iostream>
3   #include <iomanip>
4   using namespace std;
5
6   //Write your function prototype here
7   void convert (double *x)
8   int main ()
9   {//declare variable
10      double temperature;
```

```
11      //prompt user input
12      cout << "Enter a temperature in Fahrenheit, and I will
13           << convert it to Celsius: ";
14      cin >> temperature;
15      convert (&temperature) ;//calling function
16       //print the nearest decimal value by using setprecision.
17      cout << fixed << setprecision(4);
18       //print result after converting to celcious
19      cout << "Value in Celsius: " << temperature << endl;
20  return 0;
21  }
22   //Write the function convert here
23  //function
24  void convert (double *x)
    {double temp;
25  //calculation formula to Celsius
26  temp=(*x-32)*(5/9.0);
    *x = temp;
27  }
```

QUESTION 6 **[10 Marks]**

(a) Determine the output and draw a memory layout (or memory allocation) of the pointers and variables for code segment below: (7 marks)

Note: Draw a memory layout that represents C++ statement line by line.

```
int  x = 10, y = 20, z = 30;
int *ptr;

cout << x << " "  << y << " " << z << endl;
ptr = &x;
*ptr *= 10;
ptr = &y;
*ptr *= 4;
ptr = &z;
*ptr *= 2;

cout << x << " "  << y << " " << z << endl;
```

Answer:

- Memory Allocation of variables x, y, z
  - x = 4 bits
  - y = 4 bits
  - z = 4 bits

These are 4 bits because integers allocate 4 bits in the memory
How I got this?

```
int allocation = sizeof(x);
cout << allocation;
```

- Memory address of variables x, y, z
  - x = 0x61fe04
  - y = 0x61fe00
  - z = 0x61fdfc

These shows where in memory these variables are stored (in hexadecimal)
How I got this?

```
int *memoryAddress = &x;
cout << memoryAddress;
```

Explanation line by line:

1st line

```
int x = 10, y = 20, z = 30;
```

- Initialize these variables to 10, 20, 30 respectively with 4 bits memory allocation each

2nd line

```
int *ptr;
```

- create an pointer variable named ptr and allocate 4 bits into the memory because it is an integer pointer (the memory address: 0x41)

3rd line

```
cout << x << " " << y << " " << z << endl;
```

- display values of x, y, z
- Output

```
Output: 10 20 30
```

4th line

```
ptr = &x;
```

- Store the memory address of the x variable to ptr (memory address defined at the top)

5th line

```
*ptr *= 10;
```

- Dereference the ptr variable so we can change the value of the x variable.
- The operation is multiplication: x has the value of 10 then multiply it by 10
- The 5th line is the same as (*ptr = *ptr * 10)

6th line

```
ptr = &y;
```

- Store the memory address of the y variable to ptr (memory address defined at the top)

7th line

```
*ptr *= 4;
```

- Dereference the ptr variable so we can change the value of the y variable.
- The operation is multiplication: y has the value of 20 then multiply it by 4

8th line

```
ptr = &z;
```

- Store the memory address of the z variable to ptr (memory address defined at the top)

9th line

```
*ptr *= 2;
```

- Dereference the ptr variable so we can change the value of the z variable.
- The operation is multiplication: z has the value of 30 then multiply it by 2

10th line

```
cout << x << " " << y << " " << z << endl;
```

- display values of x, y, z
- Output

- `Output: 100 80 60`

Code:

```
#include<iostream>

using namespace std;

int main()

{

int       x = 10, y = 20, z = 30; int *ptr;

cout << x << " "          << y << " " << z << endl;

ptr = &x;

*ptr *= 10; ptr = &y;

*ptr *= 4; ptr = &z;

*ptr *= 2;

cout << x << " "          << y << " " << z << endl;

return 0;

}
```

Output:

```
10 20 30
100 80 60

--------------------------------
Process exited after 8.741 seconds with return value 0
Press any key to continue . . .
```

(b) Determine the output for code segment below: (3 marks)

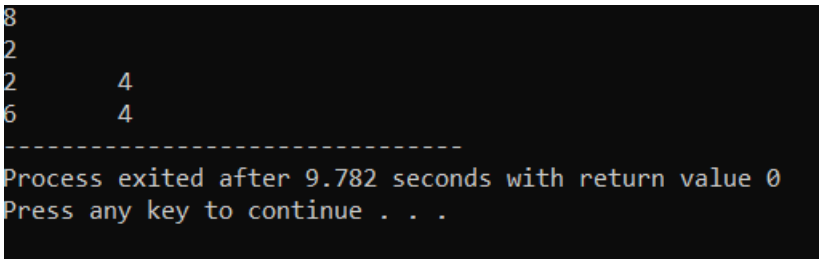
```
int numbers[5] = {2,4,6,8,10};
int *numPtr;

numPtr = numbers;
cout << *(numbers + 3) << endl;
cout << *(numbers) << endl;

for (int count=0; count < 2; count++)
{
    cout << *numPtr  << "\t";
    numPtr++;
}
cout << endl;

for (int count=0; count < 2; count++)
{
    numPtr--;
    cout << *numPtr << "\t";
}
```

Answer:

```
#include<iostream>
using namespace std;
int main()
{
int numbers[5] = {2,4,6,8,10};
int *numPtr;
numPtr = numbers;
cout << *(numbers + 3) << endl;
 cout << *(numbers) << endl;

for (int count=0; count < 2; count++)
{
cout << *numPtr    << "\t";numPtr++;

}
cout << endl;
```

```
for (int count=0; count < 2; count++)
{

cout << *numPtr << "\t";numPtr--;

}
return 0;
}
```

Output:



QUESTION 7 **[5 Marks]**

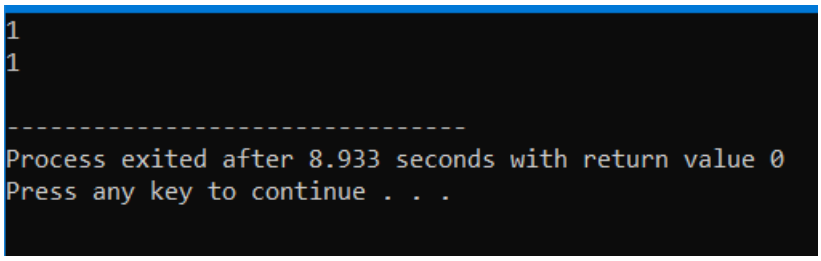
The code segment below has five (5) syntax and/ or logical errors. Identify and describe each of the errors. (5 marks)

```
struct TwoLoc
{
    int x = 10;
    int y = 15;
}

int main()
{
    TwoLoc point[10];
    point.x[0] = 1;
    point.y[0] = 1;
    cout << point << endl;
    return 0;
}
```

Answer:

```
#include <iostream>
using namespace std;

struct TwoLoc
{
  int x=10;
  int y=15;

}; //1 End of struct has to be with semicolon

int main()
{
  TwoLoc point[10];
  point[0].x=1; //2 Asigning value to a struct array using index value is correct
  point[0].y=1; //3 Asigning value to a struct array using index value is correct
  cout<<point[0].x<<endl; //4 Print the value of x using the index value is must
  cout<<point[0].y<<endl; //5 Print the value of y using the index value is must

  return 0;
}
```

Output:


1
1

Process exited after 8.933 seconds with return value 0
Press any key to continue . . .

QUESTION 8 **[16 Marks]**

Meteorological Department is responsible to keep the information about weather forecast in Malaysia and display the information through the website. Write the code segment for each of the following tasks:

(a) i) Declare a structure type named **`TempScale`**, with the following members:

`fahrenheit` : a double value

`celsius` : a double value

ii) Declare a structure type named **`Reading`**, with the following members:

`windSpeed` : an integer value

`humidity` : a double value

`temperature`: a **`TempScale`** structure variable

iii) Declare a variable of structure type **`Reading`** named **`today`**. (6 marks)

Answer:

i)
Declaration of structure type named TempScale.

```
struct TempScale
{
  double fahrenheit;
  double celsius;
};
```

ii)
Declaration of structure type named Reading

```
struct Reading
{
   int windSpeed;
   double humidity;
  TempScale temperature;
};
```

iii)

```
struct Reading today;
```

Explanation: The general syntax for declaration of structure is as follows:

```
struct <name of structure>
{
```

```
  data members; //as many we wants.
};
```
And for declaring variable of type structure is simply:
struct <name of structure> <variable name>;

(b) Write statements that will store the following data into the variable you declared in (a).

Wind speed: 8 km/h
Humidity: 53%
Fahrenheit temperature: 80.2 degrees
Celsius temperature: 26.8 degrees

Answer:

```
#include <iostream>

using namespace std;

struct Reading
{
    // To define a Reading structure to hold the readings of weather information

   double wind_speed;
   double humidity;
   double fahrenheit_temperature;
   double celsius_temperature;

};

void showReading(struct Reading r); // function prototype

int main()
{

   struct Reading r;      // PART-a) to declare a variable of Reading

   // PART-b) statements that will store the following data into the variable we have defined in PART-a)

   r.wind_speed = 8;
   r.humidity = 53;
   r.fahrenheit_temperature = 80.2;
   r.celsius_temperature = 26.8;

   showReading(r);     // Call showReading function by passing Reading structure variable 'r'
}
```

(3 marks)

(c) Define a function named **`showReading`**. It should accept a **`Reading`** structure variable as its argument. The function should display the contents of the variable onto the screen. (7 marks)

Answer:

```
/* Define a function showReading. It will accept a Reading structure variable as its
argument.This function will display the contents of the variable onto the screen */


void showReading(struct Reading r)
{

   cout << "Wind speed: " << r.wind_speed << " km/h" << endl;
   cout << "Humidity: " << r.humidity << "%" << endl;
   cout << "Fahrenheit temperature: " << r.fahrenheit_temperature << " degrees" << endl;
   cout << "Celsius temperature: " << r.celsius_temperature << " degrees" << endl;
   return 0;

}
```

PART B **[30 MARKS]**

Ministry of Transport Malaysia is required to prepare a report of the total road accidents by states in Malaysia from the year, 2006 to 2015.

INSTRUCTIONS:

Write a C++ program to calculate the average number of road accidents for each state and find the highest number of road accidents from 2006 to 2015. Your program should be able to do the following tasks:

(a) The program should use the following **`struct`** definition:

```
struct dataAcc
{
   int numAcc[10]; //number of road accidents from 2006-2015
   string state; //states in Malaysia
   float avg; //average number of road accidents for 2006-2015
}
```

(b) The program should use an array of **`struct`** defined in (a) to store the total road accidents by states in Malaysia from 2006 to 2015.

(c) The program will read input data from an input file named "input.txt" into the array of **`struct`** declared in (b). Example of input data in "input.txt" is shown in **Figure 2**.

(d) The program should have two (2) global constants and one (1) global variable as follows:

Global contants:

NUM_STATE = 14 ⇝ Number of states

NUM_YEAR = 10 ⇝ Number of years

Global variable:

out ⇝ Variable to point to output file named "output.txt"

(e) Besides the function **`main()`**, the program has three (3) other functions as described in **Table 2**. One of the functions is given below:

```
void displayLine()
{
   for (int i = 0; i < 98; i++)
```

```
        out << "-";
    out << endl;
}
```

Based on the description given in **Table 2**, the program needs to define two (2) more functions. You should use appropriate argument(s) (if necessary) for each function.

Table 2: Description of functions

Function	Description
displayLine()	To display lines using 98 characters of '-' in the output file using a loop.
cal_Avg()	To calculate the average number of road accidents for each state. The function should accept a 1D array of the number of road accidents for 10 years for each state as its argument. The function should return the average number of road accidents for each state.
find_HighLow()	To find and display the highest number of road accidents from 2006 to 2015. The function should accept a 1D array of structures as its argument.

(f) The program needs to display the following information. **Figure 3** shows the output of the successful program.

- Name of state.
- Number of road accidents for 2006-2015 in each state.
- Average number of road accidents for 10 years in each state.
- The highest number of road accidents for 10 years with the name of state and year.

The assessment criteria are shown in **Table 3**.

```
1160 1364 1417 1633 1548 1791 1881 1895 1888 1861 PERLIS
15505 16172 16520 17701 17966 19699 19935 20228 20159 22016 KEDAH
32573 33881 34049 33719 34306 37158 37851 39361 38747 39856 PULAU PINANG
27432 29203 30539 32327 32072 33506 34714 35408 35131 36736 PERAK
92632 99157 100380 107429 115565 128876 129106 135024 137809 140957 SELANGOR
46254 49454 48671 51942 53493 58795 61872 64527 63535 64664 KUALA LUMPUR
15197 16079 17362 18369 19407 21157 22146 23066 23748 22939 NEGERI SEMBILAN
10707 11720 12105 13275 14110 14720 15195 16083 16375 17069 MELAKA
43757 46584 48667 51747 55381 59501 62316 64600 64473 67112 JOHOR
13242 13982 15629 17068 17315 19001 20554 20130 19071 19635 PAHANG
7337 8116 8842 9549 9707 9603 9968 9748 10326 9960 KELANTAN
7098 8155 8814 10118 10106 10684 10861 10996 9383 10381 TERENGGANU
13550 14256 14588 15798 16192 16585 17446 17438 17858 17290 SABAH
14808 15196 15488 16655 17253 17964 18578 18700 17693 19130 SARAWAK
```

Figure 2: Input file named "**input.txt**"

STATE	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	AVERAGE
PERLIS	1160	1364	1417	1633	1548	1791	1881	1895	1888	1861	1643.8
KEDAH	15505	16172	16520	17701	17966	19699	19935	20228	20159	22016	18590.1
PULAU PINANG	32573	33881	34049	33719	34306	37158	37851	39361	38747	39856	36150.1
PERAK	27432	29203	30539	32327	32072	33506	34714	35408	35131	36736	32706.8
SELANGOR	92632	99157	100380	107429	115565	128876	129106	135024	137809	140957	118693.5
KUALA LUMPUR	46254	49454	48671	51942	53493	58795	61872	64527	63535	64664	56320.7
NEGERI SEMBILAN	15197	16079	17362	18369	19407	21157	22146	23066	23748	22939	19947.0
MELAKA	10707	11720	12105	13275	14110	14720	15195	16083	16375	17069	14135.9
JOHOR	43757	46584	48667	51747	55381	59501	62316	64600	64473	67112	56413.8
PAHANG	13242	13982	15629	17068	17315	19001	20554	20130	19071	19635	17562.7
KELANTAN	7337	8116	8842	9549	9707	9603	9968	9748	10326	9960	9315.6
TERENGGANU	7098	8155	8814	10118	10106	10684	10861	10996	9383	10381	9659.6
SABAH	13550	14256	14588	15798	16192	16585	17446	17438	17858	17290	16100.1
SARAWAK	14808	15196	15488	16655	17253	17964	18578	18700	17693	19130	17146.5

The highest number of road accidents = 140957 at SELANGOR on 2015

Figure 3: Example output in output file named "**`output.txt`**"

Table 3: Assessment criteria

Item	Criteria	Marks
A	Using an appropriate structure for the program (*e.g.* all required header files are included, the function main is properly written, *etc.*).	3
B	Define a global constants, global variable and array of structures	2
	Reading the input data from input file.	4
C	Definition of function **`cal_Avg()`**.	4
	Definition of function **`find_HighLow()`**.	7
D	Printing the output as requested in (f), Figure 3 and in a proper format.	10
Total		**30**

Answer spaces for Part B

```
#include <string>
#include <iostream>
#include <bits/stdc++.h>

using namespace std;
#define NUM_STATE 14
#define NUM_YEAR 10

ofstream out;

// structure to hold the information about accidents of each state
struct dataAcc
{
   int numAcc[10]; // number of road accidents from 2006-2015
   string state;   // states in Malaysia
   float avg;      // average number of road accidents for 2006-2015
};

typedef struct dataAcc dataAcc;

// function to display a line in the file
void displayLine()
{
   for (int i = 0; i < 98; i++)
   {
      out << "-";
   }
   out << endl;
}

// function to calculate the average of each state for accidents
float cal_Avg(int accs[])
{
   float total = 0;

   for (size_t i = 0; i < NUM_YEAR; i++)
   {
      total += accs[i];
   }

   return (total / NUM_YEAR);
}
```

```
// function to find and print high and low of the number of accidents
void find_HighLow(int accs[])
{
   // get the maximum and minimum possible for high and low initially
   int high = numeric_limits<int>::min();
   int low = numeric_limits<int>::max();

   for (int i = 0; i < NUM_YEAR; i++)
   {
      high = max(high, accs[i]);
      low = min(low, accs[i]);
   }
   cout << "highest: " << high << endl;
   cout << "lowest: " << low << endl;
}

// the main function
int main()
{
   // array of tyope dataAcc struct
   dataAcc accidents[NUM_STATE];
   ifstream in("input.txt");
   out = ofstream("output.txt");
   int index = 0;
   // if file open
   if (in.is_open())
   {
      for (index = 0; index < NUM_STATE; index++)
      {
         float total = 0;
         string state;
         in >> state;
         accidents[index].state = state;
         if (state[state.length() - 1] != ',')
         {
            in >> state;
            accidents[index].state = accidents[index].state + " " + state;
         }

         accidents[index].state = accidents[index].state.substr(0, accidents[index].state.length() - 1);

         for (int i = 0; i < 10; i++)
         {
            int numOfAcc;
            in >> numOfAcc;
```

```
            total += numOfAcc;
            accidents[index].numAcc[i] = numOfAcc;
        }
        accidents[index].avg = total / NUM_YEAR;
    }
    // cout << "file opened! " << accidents[0].state << endl;
    // cout << "file opened! " << accidents[1].state << endl;
}
// close the file
in.close();

// display the line
displayLine();
char buf[120];
sprintf(buf, "     %-13s %6d %6d %6d %6d %6d %6d %6d %6d %6d %6d  %8s ", "STATE",
2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, "AVERAGE");
out << string(buf) << endl;
displayLine();
for (size_t i = 0; i < NUM_STATE; i++)
{
    int n = accidents[i].state.length();
    char state[n + 1];
    strcpy(state, accidents[i].state.c_str());
    sprintf(buf, "%-18s %6d %6d %6d %6d %6d %6d %6d %6d %6d %6d  %8.1f ", state,
accidents[i].numAcc[0], accidents[i].numAcc[1], accidents[i].numAcc[2], accidents[i].numAcc[3],
accidents[i].numAcc[4], accidents[i].numAcc[5], accidents[i].numAcc[6], accidents[i].numAcc[7],
accidents[i].numAcc[8], accidents[i].numAcc[9], accidents[i].avg);
    out << string(buf) << endl;
}

displayLine();
string highest_state;
int highest = numeric_limits<int>::min();
int yearOffset = 0;

for (int i = 0; i < NUM_STATE; i++)
{
    for (int j = 0; j < NUM_YEAR; j++)
    {
        if (accidents[i].numAcc[j] > highest)
        {
            highest_state = accidents[i].state;
            yearOffset = j;
            highest = accidents[i].numAcc[j];
        }
```

```
        }
    }
    out << "The highest number of road accidents = " << highest << " at " << highest_state << " on "
<< 2006 + yearOffset << endl;
    displayLine();
    out.close();
    return 0;
}
```

You need to complete the assignment in group and submit before/on 2 February 2021. The answers only will affect the related items as stated in Rubric of Assignment