PART A : STRUCTURED QUESTIONS [70 marks]

1. Given a scrambled program segment that is supposed to generate a Fibonacci sequence for 10 terms (**1, 1, 2, 3, 5, 8, 13, 21, 34, 55**). However, the 4th and 7th terms are discarded from the printout as shown below. Unscramble these lines of code so that it will produce the desired output below. (8 marks)

1	`int term=1, previous=0, current=1,next;`
2	`} while (term<=10);`
3	`previous = current;`
4	`else`
5	`}`
6	`next = term;`
7	`current = next;`
8	`else`
9	`term ++;`
10	`next = previous + current;`
11	`cout << next << ", ";`
12	`{`
13	`if ((term ==5)||( term ==8))`
14	`if (term <=1)`
15	`continue;`
16	`do {`

The desired output :

```
1, 1, 2, 5, 8, 21, 34, 55,
```

Use this table to rearrange the lines according to the instructions given above.

2. a) What will the following program segment **display**, assuming the value of **n** is 5? (3 marks)

```
for (int i=0; i<n; i++)
{
            for (int j=0; j<i; j++)
                  cout<<"*";
            cout<<endl;
}
```

b) Show the **output displayed** by the following nested loops: (5 marks)

```
for (int i=0; i<2; i++)
{
      cout<<"Outer" << setw(5)<<i<<endl;
      for (int j=0; j<3; j++)
            cout<<"Inner" << setw(3)<<i<<setw(3)<<j<<endl;
      for (int k=i; k>=0; k--)
            cout<<"Inner" << setw(3)<<i<<setw(3)<<k<<endl;
}
```

3. Using nested **`for`** loops only, write the code segment that will print each of the following output:

(4 marks)

a)

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

b)

```
1 1 1 1 1
2 2 2 2
3 3 3
4 4
5
```

4. Based on the C++ program in **`Program 1`**, trace the following set of inputs:

   (i) **3 4 2 y**
   (ii) **5 3 7 Y**
   (iii) **0 9 3 n**

(5 marks)

```
//Program 1
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    char ans='y';
    int x, count, i, n=3;

    while (ans=='y'|| ans=='Y')
    {
      count=0;
      i=0;

      while(i<n)
      {
       cout<<"\nEnter value for x: ";
       cin>>x;

       if (x==i)
       {
        ++count;
        cout<<"Count : "<<count<<endl;
       }

       ++i;
       cout<<"i : "<<i<<endl;
      } //end-while

      cout<<"\nDo you want to continue? Y-yes or N-No :";
      cin>>ans;

    }//end-while

    return 0;
}
```

Trace Table:

Set of inputs	**x**	**i**	**count**	**ans**
(i)				
(ii)				
(iii)				

5. a) What is the output of **Program 2** below? (3 marks)

```
//Program 2
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

int main()
{
      int n1, n2, n3,x=4 ;
      n1 = pow(2,3);
      n2  = sqrt(x+sqrt(25));
      n3  = n1 % n2 ;
      cout << " value 1 = " << n1 << endl;
      cout << " value 2 = " << n2 << endl;
     cout << " value 3 = " << n3 ;
   return 0;
}
```

b) What is the output of **Program 3** below? (4 marks)

```
//Program 3
#include <iostream>
using namespace std;
int f(int,int);

int main()
{
     int a= 1, b =5,c,p;
     c = f(a,b);
     cout  <<  c  << endl;
     p =   f( c,3);
     cout  <<  p << endl;
     return 0;
}

int  f(int   x,  int  y)
{
     int  m,n,k ;
     m = x + 2;
     n  = y - 1 ;
     k = m + n;
     cout   << m  << "   "  << n << endl;
     return  k;
}
```

6. **Program 4** is to calculate the smallest and the average of 3 numbers. Complete **Program 4** below based on the questions in (a) and (b).

```
//Program 4
#include <iostream>
using namespace std;
void smallest(int,int,int) ;
void largest(int,int,int);
int ave( int,int,int);

int  main()
{  int x,y,z,average;

   cout << "Key in 3 integer numbers : ";
   cin >> x  >> y   >> z;
   smallest(x,y,z) ;
   average = ave(x,y,z);
   cout  << "Average is " << average ;

   return 0;   }
```

(a) Write a function named **smallest()** that will find and display the value of the smallest number.

(4 marks)

(b) Write a function named **ave()** to get the average value of 3 numbers. (2 marks)

7. **`swapOdd`** and **`countOddEven`** are functions that pass parameters by reference. Define the functions according to the description in (a) and (b). For each function write a statement which shows how to call the function.

   a) **`swapOdd`**. This function receives two numbers and swaps these numbers only if both numbers are odd. (4.5 marks)

   b) **`countOddEven`**. This function receives two numbers, ***m*** and ***n***, and counts the number of odd and even numbers from ***m*** to ***n***. For example, if *m* is 1 and *n* is 5, then the count of odd and even numbers are 3 and 2, respectively. (7.5 marks)

8. A palindrome is a word which is identical regardless whether it is read from the front or from the back. From an input which is an array of 5 characters, determine if it is a palindrome. Examples of palindromes are words such as MADAM and ROTOR. Words which are **not** palindromes are COLOR and MOTOR.

   Complete **`Program 5`** below to identify whether a word is a palindrome or otherwise. Finally print the word and its status, for example:

```
MADAM is a palindrome.
ROTOR is a palindrome.
COLOR is not a palindrome.
MOTOR is not a palindrome.
```

(9 marks)

```
//Program 5
#include <iostream>
using namespace std;

int main()
{     // declare variables and/or initialize variables (2 marks)
      _____________________________;
      _____________________________;
      _____________________________;
      _____________________________;
      cout << "Enter a 5 letter word ";
      _____________________________;  // (1 mark)
      // set the length of the word to a variable
      _____________________________;  // (0.5 mark)
      // loop to test if word entered is a palindrome
      ___________________________________________   // (1.5 marks)
     {  // compare respective characters in the word to determine palindrome status
            ___________________________________   // (1.5 marks)
                        ___________________; }
      // test to print status: either the word is a palindrome or otherwise
      ___________________   // (0.5 mark)
            ___________________________________________;  // (1 mark)
      else
            ___________________________________________;  // (1 mark)

      return 0;  }
```

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

```
string students[] = {"Ali", "Chong", "Muthu", "Salmah"};
int midTestMark[] = {75, 80, 81, 88};
int totalMark[4];

int SBtest[4][3]  = { {10, 7, 6}, {6, 7, 8}, {9, 7, 8},
                      {10, 6, 7}  };
```

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

a) Write **a code segment** that prints the student names along with their Mid Term Test Marks. The output should be as follows: (3 Marks)

```
Ali     75
Chong   80
Muthu   81
Salmah  88
```

b) Write **a code segment** that produces the following output from the array named **`table`**.

(4 marks)

```
Name       SBT1  SBT2  SBT3
Ali        10    7     6
Chong      6     7     8
Muthu      9     7     8
Salmah      10   6     7
```

(more answering space on the next page)

c) Given the following function, write a **code segment** to calculate the total mark of the student. The codes also store the total mark value in the array **`totalMark`** and print the list of the students' marks as follows. (4 marks)

```
int averageSBT(const int m[])
{
      int total=0;
    for (int i=0; i<3; i++)
        total += m[i];
    return total/3;
}
```

```
Mid-Term Test and SBT Marks
Ali        82
Chong      87
Muthu      89
Salmah     95
```

PART B : PROGRAMMING QUESTION [30 marks]

A tele match event has been held in Sekolah Rendah Tebing Tinggi. Three teams are allowed to participate in this match, with each team consisting of four participants. Five (5) events were contested, namely E1, E2, E3, E4 and E5. Table 1 shows the scores that have been collected by each team for the five events.

Table 1: Collected scores

Team ID	Participant ID	E1	E2	E3	E4	E5
1	1001	10	5	8	10	6
	1002	8	7	10	7	9
	1003	7	10	10	6	10
	1004	10	10	8	7	7
2	2001	7	8	10	9	10
	2002	10	8	7	8	10
	2003	8	6	8	8	10
	2004	7	8	8	8	8
3	3001	10	9	10	10	10
	3002	8	7	8	8	8
	3003	7	8	9	10	6
	3004	8	6	8	7	7

INSTRUCTIONS:

Write a C++ program which can assist the tele match committee to determine the winner for these events. Your program should be able to do the following tasks:

(a) The program will read input data: team ID, participant ID and **scores** for the five events namely E1, E2, E3, E4 and E5 from an input file named **`"input.txt"`** into an array **`marks[12][7]`** of type **`int`**. Example of the series of input data in input file is shown in Figure 1.

(b) The program must be able to notify the user if the input file cannot be opened (failed to open) with proper prompt. The example for user notification where the file fails to open is shown in Figure 2.

(c) Calculate the total score for each participant.

(d) Calculate the total score for each team.

(e) Besides the function **`main()`**, the program needs to define **three** (3) other functions as described in Table 2. Use appropriate argument (if necessary) for each function.

(f) The program needs to print out the following information. Figure 3 shows the example, run of the successful program.

i. The team ID.
ii. The participant ID.
iii. The **scores** for the five events, E1, E2, E3, E4 and E5 for each participant.
iv. The total score for each participant.
v. The total score for each team.
vi. The winner for individual category (selected based on highest total score that collected by the participants).
vii. The winner for group category (selected based on highest total score that collected by the teams).

```
1 1001 10 5 8 10 6
1 1002 8 7 10 7 9
1 1003 7 10 10 6 10
1 1004 10 10 8 7 7
2 2001 7 8 10 9 10
2 2002 10 8 7 8 10
2 2003 8 6 8 8 10
2 2004 7 8 8 8 8
3 3001 10 9 10 10 10
3 3002 8 7 8 8 8
3 3003 7 8 9 10 6
3 3004 8 6 8 7
```

Figure 1: Input file named "**`input.txt`**"

```
Sorry, input file does not exist!
Press any key to continue . . .
```

Figure 2: Example user notification in case file fails to open

Table 2: Description for functions

Function	Description
`displayLine()`	To display lines using the 52 characters of '-'. The function should use loop to display the line.
`findIndWinner()`	To determine the winner for individual category (selected based on the highest total score that was collected by participants). The function should accept the array for a total score of each participant as one of its argument.
`findTeamWinner()`	To determine the winner of group category (selected based on the highest total score that was collected by teams). The function should accept the array for a total score for each team as one of its argument.

```
-----------------------------------------------------
Id      E1      E2      E3      E4      E5      Total
-----------------------------------------------------
TEAM 1
1001    10      5       8       10      6       39
1002    8       7       10      7       9       41
1003    7       10      10      6       10      43
1004    10      10      8       7       7       42
TOTAL                                           165
-----------------------------------------------------
TEAM 2
2001    7       8       10      9       10      44
2002    10      8       7       8       10      43
2003    8       6       8       8       10      40
2004    7       8       8       8       8       39
TOTAL                                           166
-----------------------------------------------------
TEAM 3
3001    10      9       10      10      10      49
3002    8       7       8       8       8       39
3003    7       8       9       10      6       40
3004    8       6       8       7       7       36
TOTAL                                           164
-----------------------------------------------------

Winner for Individual Category: 3001 (Team 3)
Winner for Group Category: Team 2 (Score = 166)

Press any key to continue . . .
```

Figure 3: Example run of the program

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

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.*)	1
B	Determining either the input file can be opened or not, and notify the user if the input file cannot be opened (failed to open) with proper message.	2
	Reading the input data: team ID, participant ID and **scores** for the five events namely E1, E2, E3, E4 and E5 from an input file.	1.5
C	Calculating the total score for each participant.	2
	Calculating the total score for each team.	2.5
D	Declaration of function `displayLine()`, `findIndWinner()` and `findTeamWinner()`.	2
	Code for the function displayLine `()`: displaying lines using the 52 characters of '-'. The function should use loop to display the line.	2
	Code for the function `findIndWinner()`: determining the winner for individual category (selected based on the highest total score that collected by participants). The function should accept the array for a total score of each participant as one of its argument.	3.5
	Code for the function `findTeamWinner()`: determining the winner of group category (selected based on the highest total score that collected by teams). The function should accept the array for a total score for each team as one of its argument.	3.5
E	Printing the team ID, participant ID, and **scores** for the five events, E1, E2, E3, E4 and E5 for each participant.	1.5
	Printing the total score for each participant.	0.5
	Printing the total score for each team.	1
	Printing the winner for individual category (selected based on highest total score that collected by the participants).	1.5
	Printing the winner for group category (selected based on highest total score that collected by the teams).	1.5
	Printing the output in a proper format.	2
F	Defining one file stream object for input file.	0.5
	Opening input file object.	1
	Closing input file object.	0.5
Total		**30**

-- THIS PAGE IS INTENTIONALLY LEFT BLANK FOR ANSWERS --

(please request for more papers for answering this question)