SOLUTIONS

PART A [70 marks]

Question 1

1. (a)
```
char students[5][30] = {"Amir Ali", "Rania", "Syamim", "Thomas
Ang", "Aidura"};                                              (2 marks)
```

(b)
```
for (int i = 0; i < 5; i++)
   cout << students[i] << endl;                               (2 marks)
```

(c)
```
void changeName (char stud[5][30])
{
   for (int i = 0; i < 5; i++)
      cin.getline (stud[i], 30);
}                                                             (4 marks)
```

(d)
```
changeName(students);                                         (1 mark)
```

Alternative answer:

(a)
```
string students[5] = {"Amir Ali", "Rania", "Syamim", "Thomas
Ang", "Aidura"};                                              (2 marks)
```

(b)
```
for (int i = 0; i < 5; i++)
   cout << students[i] << endl;                               (2 marks)
```

(c)
```
void changeName (string stud[5])
{
   for (int i = 0; i < 5; i++)
      getline (cin, stud[i]);
}                                                             (4 marks)
```

(d)
```
changeName(students);                                         (1 mark)
```

Question 2 (4 Marks)

0.5 m for each value. 0.5m x 2 x 4

		Answers	
		x	***y***
i.	`f(x, y);`	**3**	**2**
ii.	`y = f(x, 2);`	**3**	**4**
iii.	`y = f(x);`	**2**	**4**
iv.	`y = f (x, f(2) );`	**3**	**6**

Question 3 (12 Marks)

0.5 m for each value. 0.5m x 2 x 4

a. (7 marks)

```
// Function header (0.5m)
int readPositive()
{   int n;
    do{
          // read input (0.5m)
          cout << "Enter an integer => ";
          cin >> n;

    } while (n<=0);
    // The loop, make sure the value is positive,
    //  otherwise, asks the user to enter another
    //   value (0.5m)

    // Return the number to the caller  (0.5m)
    return n;
}
```

2 marks

```
// Function header (0.5m)

void swap(int &x, int &y)

{    // Swapping process (1.5m)

     int z = x;

     x = y;

     y = z;

}
```

2 marks

```
// Function header (0.5m)

int multiply(int x, int y)

{    int p=1;   // Initilize p. (0.5m)


     // Do the multiplication ascendingly

     for (int n=x; n<=y; n++)   // (1m)

            p = p * n;            // (0.5m)


     // Return the result to the caller  (0.5m)

     return p;

}
```

3marks

b (5 marks)

```
int main()

{

      int s;  // The starting integer number.

      int e;  // The ending integer number.

      int p;  // The product of the integers from s to e.


      //Task 1: Read the starting integer number. It must be positive.  (1 mark)

      cout <<"The starting number:" << endl;
```

```
        s = readPositive();


        cout << endl;

        //Task 2: Read the ending integer number.  It must be positive.  (1 mark)

        cout <<"The ending number:" << endl;

        e = readPositive();


        // Task 3: Make sure that  s ≤ e, otherwise swap their values. (2 marks)

        if (s>e)

              swap(s,e);

        // Task 4: Find the product of integer numbers from s  to e.   (1 mark)

        p = multiply(s,e);


        cout << endl;

        cout << "The product of the integer numbers is "

             << p << endl;


        return 0;

}
```

Question 4 (1 mark for each answer)

Line	C++ Statements	Output
1	#include <iostream>	
2	#include <iomanip>	
3	#include <cmath>	
4	using namespace std;	
5	int main() {	
6	float num1 = -6.25;	
7	cout << showpoint << fixed;	
8	cout << fabs(num1) << endl;	6.250000
9	cout << ceil(num1) << endl;	-6.000000
10	cout << floor(num1) << endl;	-7.000000
11	cout << pow(num1, 2) << endl;	39.062500

12	int num2 = abs(static_cast<int>(num1));
	@ int num2 = abs((int)num1);	

13	cout << num2 << endl;	6
14	cout << sqrt(num2) << endl;	2.449490
15	cout << log10(num2) << endl;	0.778151
16	return 0; }	

Question 5 (18 Marks)

a. (6 marks). 0.5 m each number

i	33 34 35 36 37	2.5 marks
ii	13 27 35 43	2 marks
iii	11 25 26	1.5 marks

b. (12 marks)

// (i) print the sum all the elements of p (5m)

// 1m each line

```
int sum = 0;
for (int i=0; i<4; i++)
    for (int j=0; j<5; j++)
        sum += p[i][j];
cout<< "Sum = " << sum << endl << endl;
```

// (ii) copy p to q (3m)

// 1m each line

```
for (int i=0; i<4; i++)
    for (int j=0; j<5; j++)
        q[i][j] = p[i][j];
```

// (iii) find the total of elements for each column of array p. Put the result into array r (4m)

// 1m each line

```
for (int j=0; j<5; j++){
    r[j] = 0;
    for (int i=0; i<4; i++)
        r[j] += p[i][j];
}
```

Question 6

(a). Total – 5 marks

```
struct Tenants    {        // struct-0.5m, Tenants-0.5m
     int lotNumber;         // the rest 0.5m each
     char lotStatus;
     char tenantName[30];
     char phoneNum[11];
     double lotLength;
     double lotWidth;
     double rentalRate;
     double mthlyRent;

};
```

(b). Total – 3 marks

```
Tenants tenant[3] = {{2317,'P',"Shafie
Afdal","017323477",12,15},                                    1m
                        {1224,'O',"Linda                      each
                  Malek","0113211212",5,12}};

     0.5m each
```

(c). Total – 3 marks

```
        cin>>tenant[2].lotNumber;   //0.5m each
        cin>>tenant[2].lotStatus;
        cin>>tenant[2].tenantName;
        cin>>tenant[2].phoneNum;
        cin>>tenant[2].lotLength;

        cin>>tenant[2].lotWidth;
```

(d). Total – 5 marks

```
     for (int i=0; i<3; i++) // 1m each for/assignment/condition
0.5m
each   {  if (tenant[i].lotStatus=='O')
for if       tenant[i].rentalRate = 100;
& else    else
             tenant[i].rentalRate = 100 * 2;    }
```

(e). Total – 3 marks

```
     for (int i=0; i<3; i++) // 1m each for/assignment/condition

     {  tenant[i].mthlyRent =
           (tenant[i].lotLength * tenant[i].lotWidth) *
              tenant[i].rentalRate;    }
```

PART B **[30 marks]**

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

#define MAX 5
#define REG_LIMIT 40
#define OT_FACTOR 1.5
#define HOURS 0
#define RATE 1
#define REGPAY 2
#define OVER 3
#define TOTAL 4

//function prototype 5 markah
void displayLine();
void getData (int id[], double payrec[][TOTAL+1]);
void calcPay(double payrec[][TOTAL+1]);
void printData(int id[], double payrec[][TOTAL+1]);
void highestOvertime(int ID[], double payrec[][TOTAL+1]);

main(){
      int id[MAX];  double payrec[MAX][5];

     cout<<"Payroll Program"<<endl;
     getData(id,payrec);
     calcPay(payrec);
     printData(id,payrec);
     highestOvertime(id, payrec);
}
void displayLine()//2M
{
  for (int j = 0; j <= 105; j++)
    cout << "-";
  cout << endl;
}

void getData(int id[], double payrec[][TOTAL+1])
{
     int counter=0;
     float h, rate;

   while (counter<MAX)
   {
     cout<<"ID: ";
     cin>>id[counter];
     cout<<"Hours worked:";
     cin>>h;
     payrec[counter][HOURS]=h;
     cout<<"Rate of Pay (RM per hour):";
     cin>>rate;
     payrec[counter][RATE]=rate;
     cout<<"\t"<<id[counter]<<"\t"<<payrec[counter][HOURS]<<"\
     t"<<payrec[counter][RATE]<<endl;
     counter++;

   }
}

   void calcPay(double payrec[][TOTAL+1])
   {
       for (int i=0; i<MAX; i++){

          if (payrec[i][HOURS]<=REG_LIMIT){

               payrec[i][REGPAY]                              =
               payrec[i][HOURS]*payrec[i][RATE];
               payrec[i][OVER]=0;
           }
         else{

               payrec[i][REGPAY] = REG_LIMIT *payrec[i][RATE];
               payrec[i][OVER]= (payrec[i][HOURS] - REG_LIMIT)
               * OT_FACTOR *  payrec[i][RATE];
          }

          payrec[i][TOTAL]=payrec[i][REGPAY]                  +
          payrec[i][OVER];

     cout<<"\t"<<payrec[i][HOURS]<<"\t"<<payrec[i][RATE]<<"\t"
     <<payrec[i][REGPAY]
     <<"\t"<<payrec[i][OVER]<<"\t"<<payrec[i][TOTAL]<<endl;
      }

   }

void printData(int ID[], double payrec[][TOTAL+1]){
     int i,j;
     ofstream out;

     out.open("payroll.txt");
     cout<<"Payroll Final Report "<<endl;
     out<<"Payroll Final Report "<<endl;
     displayLine();

     cout<<setw(5)<<"ID"<<setw(10)<<"HOURS"<<setw(15)<<"RATE(R
     M)"<<setw(20)<<"REGULAR   PAY   (RM)"<<setw(25)<<"OVERTIME
     (RM)"<<setw(30)<<"TOTAL (RM)"<<endl;
     displayLine();


     for (i=0; i<MAX; i++){

          cout<<setw(5)<<ID[i];
          out<<setw(5)<<ID[i];

          cout<<setw(10)<<payrec[i][HOURS]<<setw(15)<<payrec[i
          ][RATE]<<setw(20)<<payrec[i][REGPAY]
          <<setw(25)<<payrec[i][OVER]<<setw(30)<<payrec[i][TOT
          AL]<<endl;

    }

     out.close();

}
void highestOvertime(int ID[], double payrec[][TOTAL+1])
//3.5M
{
     int staffID = ID[0];
     double overtime= payrec[0][OVER];

     cout<<staffID<<endl;
     cout<<overtime<<endl;


     for (int i = 0; i<MAX; i++){
          if (payrec[i][OVER]>=overtime)
          {
                    staffID = ID[i];
                    overtime = payrec[i][OVER];
              }
    }
    cout<<"Staff "<<staffID<<" have the highest pay overtime
    of RM"<<overtime;
}
```