Question 1

Write a program that requires users to key-in their tuition fee. The program will calculate the total fees for the month and the average payment to be made. Create a dynamic array with the *new* operator to store the elements entered by user.

[HINT: Complete the given program by referring to the sample output]

```
#include <iostream>
using namespace std;
int main ( )
{   int subject, num;
    float total_payment, average_payment;
    float *fees;

    __________          /* Ask user for the number
    __________          of subjects registered */

    fees= new float[subject];
    for (num=0; num<subject; num++)

    __________          /* Ask user for the fees charged for
    __________          each class registered. Then,
    __________          calculate the total payment */
    __________
    __________          /* Display the details of the fees
    __________          entered earlier */
    __________          /* Calculate the average payment,
    __________          then display the total payment &
    __________          minimum payment (which is the
    __________          average).*/

    return 0;
}
```

Sample Output Screen

```
How many subjects you have registered: 3
Class 1 fees charge: RM 60
Class 2 fees charge: RM 50
Class 3 fees charge: Rm 70

===== PAYMENT DETAILS =====
Fees charges you have entered: 60(RM)... 50(RM)...70(RM)...
Total payment to be made: RM 180
Minimum payment (Average of total payment): RM 60
```

Question 2

Write a C++ program that contains:

a) A constant global variable ***ITEM*** with value **3**.

b) A class ***BeanBag*** with the following:

   (i) Private data members: **code *name* (string); *stock [ITEM]* (int)**

   (ii) Public member functions:

   - **latest_BeanBagStock ( )**
     - Display the ***"#Current# Ready Stocks……."*** with the code *name* and the array elements of *stock* in reverse order using an appropriate looping structure.
   - **beanBag_Details (int *)**
     - This function prints details of ***"Stocks Checking"***
     - Gets user input for the code *name.*
     - The function has a pointer argument.
     - In a *for* loop, use the pointer argument to initialize the *stock* array.
   - A global object declaration, named ***ready.***

c) A function named **stockUpdate()**:

   (i) Refer to label '***stockUpdate ( )***' at sample output.

   (ii) Get user input for 3 values that should be stored in a dynamic array created with the *new* operator.

   (iii) Using the global object ***ready***, call the function ***beanBag_Details(…)***, passing in the array and also call ***latest_BeanBagStock()*** after that.

   (iv) Delete the dynamic array created.

d) In the *main()*:

   (i) Declare an object of the class BeanBag

   (ii) Declare an array of 3 integer elements and initialize it with the values {9,7,5}

   (iii) Using the object (created at d)(i)), call ***beanBag_Details (….)*** with the array declared at d)(ii).

   (iv) Using the object (created at d)(i)), call ***latest_BeanBagStock ()***.

   (v) Call ***stockUpdate()***

Sample Output Screen

```
COSY Bean Bag Chair Sdn. Bhd.
#######################################
        stocks Checking
---------------------------------------
Enter Bean Bag Chair code: AB102
--------------------------------------
#Current# Ready stocks Checking...
--------------------------------------
Code Tracing    >>AB102<<
Group 1 production:     5 item(s) ready
Group 2 production:     7 item(s) ready
Group 3 production:     9 item(s) ready
```

```
------------------------------------------
 Ready stocks for this Month
------------------------------------------
Ready stock from group 3:90
Ready stock from group 2:54
Ready stock from group 1:71
#######################################
        stocks Checking
---------------------------------------
Enter Bean Bag Chair code: BQ103
--------------------------------------
#Current# Ready stocks Checking...
--------------------------------------
Code Tracing    >>BQ103<<
Group 1 production:     71 item(s) ready
Group 2 production:     54 item(s) ready
Group 3 production:     90 item(s) ready
------------------------------------------
```

stockUpdate ( )