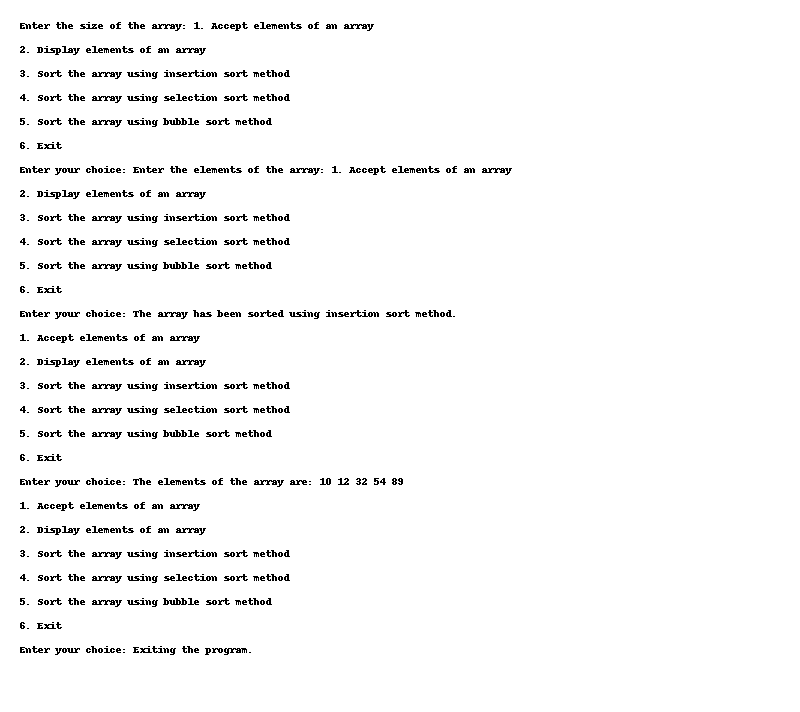
# Code: 3-1.cpp

/\*  
Write a menu driven C++ program with following option  
a. Accept elements of an array  
b. Display elements of an array  
c. Sort the array using insertion sort method  
d. Sort the array using selection sort method  
e. Sort the array using bubble sort method  
\*/  
  
#include <iostream>  
using namespace std;  
  
void accept(int arr[], int n)  
{  
 cout << "Enter the elements of the array: ";  
 for (int i = 0; i < n; i++)  
 {  
 cin >> arr[i];  
 }  
}  
  
void display(int arr[], int n)  
{  
 cout << "The elements of the array are: ";  
 for (int i = 0; i < n; i++)  
 {  
 cout << arr[i] << " ";  
 }  
 cout << endl;  
}  
  
void insertionSort(int arr[], int n)  
{  
 int key, j;  
 for (int i = 1; i < n; i++)  
 {  
 key = arr[i];  
 j = i - 1;  
 while (j >= 0 && arr[j] > key)  
 {  
 arr[j + 1] = arr[j];  
 j = j - 1;  
 }  
 arr[j + 1] = key;  
 }  
}  
  
void selectionSort(int arr[], int n)  
{  
 int i, j, minIndex;  
 for (i = 0; i < n - 1; i++)  
 {  
 minIndex = i;  
 for (j = i + 1; j < n; j++)  
 {  
 if (arr[j] < arr[minIndex])  
 {  
 minIndex = j;  
 }  
 }  
 swap(arr[minIndex], arr[i]);  
 }  
}  
  
void bubbleSort(int arr[], int n)  
{  
 for (int i = 0; i < n - 1; i++)  
 {  
 for (int j = 0; j < n - i - 1; j++)  
 {  
 if (arr[j] > arr[j + 1])  
 {  
 swap(arr[j], arr[j + 1]);  
 }  
 }  
 }  
}  
  
int main()  
{  
 int n, choice;  
 cout << "Enter the size of the array: ";  
 cin >> n;  
 int arr[n];  
 do  
 {  
 cout << "1. Accept elements of an array" << endl;  
 cout << "2. Display elements of an array" << endl;  
 cout << "3. Sort the array using insertion sort method" << endl;  
 cout << "4. Sort the array using selection sort method" << endl;  
 cout << "5. Sort the array using bubble sort method" << endl;  
 cout << "6. Exit" << endl;  
 cout << "Enter your choice: ";  
 cin >> choice;  
 switch (choice)  
 {  
 case 1:  
 accept(arr, n);  
 break;  
 case 2:  
 display(arr, n);  
 break;  
 case 3:  
 insertionSort(arr, n);  
 cout << "The array has been sorted using insertion sort method." << endl;  
 break;  
 case 4:  
 selectionSort(arr, n);  
 cout << "The array has been sorted using selection sort method." << endl;  
 break;  
 case 5:  
 bubbleSort(arr, n);  
 cout << "The array has been sorted using bubble sort method." << endl;  
 break;  
 case 6:  
 cout << "Exiting the program." << endl;  
 break;  
 default:  
 cout << "Invalid choice." << endl;  
 }  
 } while (choice != 6);  
 return 0;  
}

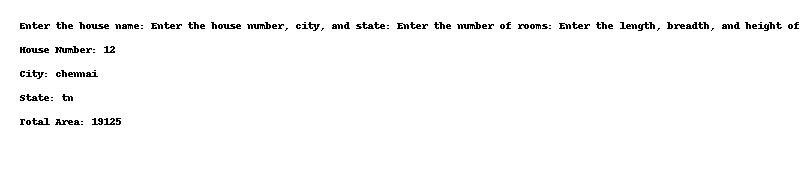
## Output



# Code: 3-10.cpp

/\*  
Tamilnadu land registration authority is panning to keep track of the native addresses and total area of  
the flats people have across the state. Since the total population and area need to be monitored is huge.  
Government is looking for the software which does this task. Can you help them with proper  
programming logic for implementing the same?  
Constraints:  
1â‰¤ hno<500  
1< no room< 10  
1â‰¤ length < 50  
1< breadth < 50  
1â‰¤ height < 50  
Input Format:  
The first line of the input contain a single string denoting the house name.  
The second line of the input contain three values of type Integer String and String separated  
by a space representing house number, city and state respectively. The third line of the input has a single Integer representing the number of rooms.  
The subsequent lines of input must have length, breadth and height of each room  
Output Format:  
Print the details of the house in the expected format.  
\*/  
#include <iostream>  
using namespace std;  
  
class House  
{  
 private:  
 string houseName;  
 int houseNumber;  
 string city;  
 string state;  
 int numberOfRooms;  
 int totalArea;  
 public:  
 House(string hn, int hno, string c, string s, int nor)  
 {  
 houseName = hn;  
 houseNumber = hno;  
 city = c;  
 state = s;  
 numberOfRooms = nor;  
 totalArea = 0;  
 }  
 void getRoomDetails()  
 {  
 int length, breadth, height;  
 for (int i = 0; i < numberOfRooms; i++)  
 {  
 cout << "Enter the length, breadth, and height of room " << i + 1 << ": ";  
 cin >> length >> breadth >> height;  
 totalArea += length \* breadth \* height;  
 }  
 }  
 void display()  
 {  
 cout << "House Name: " << houseName << endl;  
 cout << "House Number: " << houseNumber << endl;  
 cout << "City: " << city << endl;  
 cout << "State: " << state << endl;  
 cout << "Total Area: " << totalArea << endl;  
 }  
};  
  
int main()  
{  
 string houseName, city, state;  
 int houseNumber, numberOfRooms;  
 cout << "Enter the house name: ";  
 cin >> houseName;  
 cout << "Enter the house number, city, and state: ";  
 cin >> houseNumber >> city >> state;  
 cout << "Enter the number of rooms: ";  
 cin >> numberOfRooms;  
 House obj(houseName, houseNumber, city, state, numberOfRooms);  
 obj.getRoomDetails();  
 obj.display();  
 return 0;  
}

## Output



# Code: 3-2.cpp

/\*  
Write a C++ program to add two numbers using the concept of data abstraction  
\*/  
  
#include <iostream>  
using namespace std;  
  
class Add  
{  
 private:  
 int a, b;  
 public:  
 void get()  
 {  
 cout << "Enter two numbers: ";  
 cin >> a >> b;  
 }  
 void display()  
 {  
 cout << "The sum of " << a << " and " << b << " is " << a + b << endl;  
 }  
};  
  
int main()  
{  
 Add obj;  
 obj.get();  
 obj.display();  
 return 0;  
}

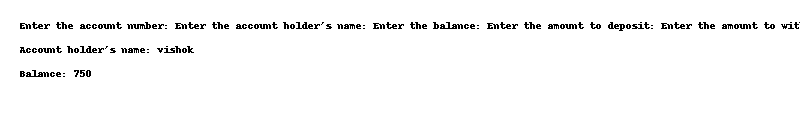
## Output



# Code: 3-3.cpp

/\*  
Develop a class to represent a bank account. Include methods to deposit money, withdraw money, and  
check the account balance. Ensure that the withdrawal method checks for sufficient funds.  
\*/  
#include <iostream>  
using namespace std;  
  
class BankAccount  
{  
 private:  
 int accountNumber;  
 string accountHolderName;  
 float balance;  
 public:  
 void get()  
 {  
 cout << "Enter the account number: ";  
 cin >> accountNumber;  
 cout << "Enter the account holder's name: ";  
 cin >> accountHolderName;  
 cout << "Enter the balance: ";  
 cin >> balance;  
 }  
 void deposit()  
 {  
 float amount;  
 cout << "Enter the amount to deposit: ";  
 cin >> amount;  
 balance += amount;  
 }  
 void withdraw()  
 {  
 float amount;  
 cout << "Enter the amount to withdraw: ";  
 cin >> amount;  
 if (amount > balance)  
 {  
 cout << "Insufficient funds" << endl;  
 }  
 else  
 {  
 balance -= amount;  
 }  
 }  
 void display()  
 {  
 cout << "Account number: " << accountNumber << endl;  
 cout << "Account holder's name: " << accountHolderName << endl;  
 cout << "Balance: " << balance << endl;  
 }  
};  
  
int main()  
{  
 BankAccount obj;  
 obj.get();  
 obj.deposit();  
 obj.withdraw();  
 obj.display();  
 return 0;  
}

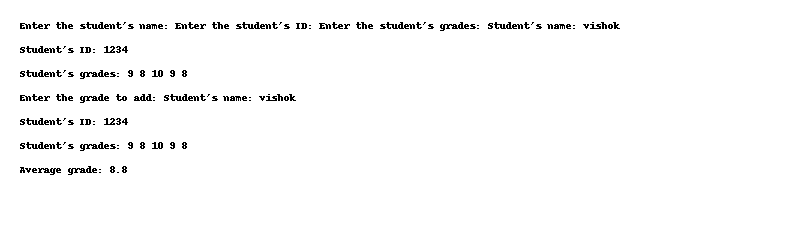
## Output



# Code: 3-4.cpp

/\*  
Create a class to represent a student in a grading system. Include attributes for the student's name, ID,  
and an array to store grades. Implement methods to calculate the average grade, display the student's  
information, and add a new grade.  
\*/  
#include <iostream>  
using namespace std;  
  
class Student  
{  
 private:  
 string name;  
 int id;  
 int grades[5];  
 public:  
 void get()  
 {  
 cout << "Enter the student's name: ";  
 cin >> name;  
 cout << "Enter the student's ID: ";  
 cin >> id;  
 cout << "Enter the student's grades: ";  
 for (int i = 0; i < 5; i++)  
 {  
 cin >> grades[i];  
 }  
 }  
 void display()  
 {  
 cout << "Student's name: " << name << endl;  
 cout << "Student's ID: " << id << endl;  
 cout << "Student's grades: ";  
 for (int i = 0; i < 5; i++)  
 {  
 cout << grades[i] << " ";  
 }  
 cout << endl;  
 }  
 void addGrade()  
 {  
 int grade;  
 cout << "Enter the grade to add: ";  
 cin >> grade;  
 for (int i = 0; i < 5; i++)  
 {  
 if (grades[i] == 0)  
 {  
 grades[i] = grade;  
 break;  
 }  
 }  
 }  
 void averageGrade()  
 {  
 float sum = 0;  
 for (int i = 0; i < 5; i++)  
 {  
 sum += grades[i];  
 }  
 cout << "Average grade: " << sum / 5 << endl;  
 }  
};  
  
int main()  
{  
 Student obj;  
 obj.get();  
 obj.display();  
 obj.addGrade();  
 obj.display();  
 obj.averageGrade();  
 return 0;  
}

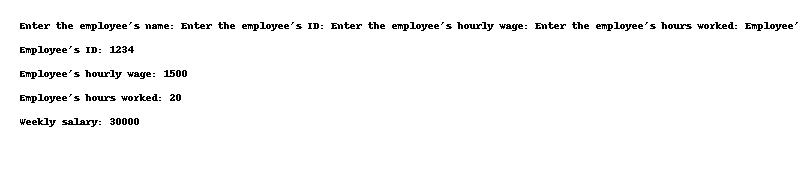
## Output



# Code: 3-5.cpp

/\*  
Develop a class to represent an employee in a payroll system. Include attributes for the employee's  
name, ID, hourly wage, and hours worked. Implement methods to calculate the weekly salary and  
display the employee's information.  
\*/  
#include <iostream>  
using namespace std;  
  
class Employee  
{  
 private:  
 string name;  
 int id;  
 float hourlyWage;  
 float hoursWorked;  
 public:  
 void get()  
 {  
 cout << "Enter the employee's name: ";  
 cin >> name;  
 cout << "Enter the employee's ID: ";  
 cin >> id;  
 cout << "Enter the employee's hourly wage: ";  
 cin >> hourlyWage;  
 cout << "Enter the employee's hours worked: ";  
 cin >> hoursWorked;  
 }  
 void display()  
 {  
 cout << "Employee's name: " << name << endl;  
 cout << "Employee's ID: " << id << endl;  
 cout << "Employee's hourly wage: " << hourlyWage << endl;  
 cout << "Employee's hours worked: " << hoursWorked << endl;  
 }  
 void weeklySalary()  
 {  
 cout << "Weekly salary: " << hourlyWage \* hoursWorked << endl;  
 }  
};  
  
int main()  
{  
 Employee obj;  
 obj.get();  
 obj.display();  
 obj.weeklySalary();  
 return 0;  
}

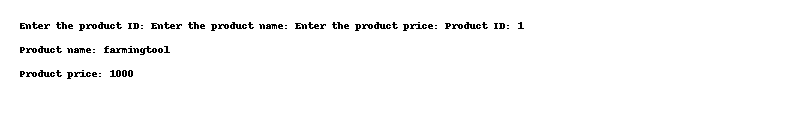
## Output



# Code: 3-6.cpp

/\*  
Write a C++ program to display product detail using classes.  
\*/  
  
#include <iostream>  
using namespace std;  
  
class Product  
{  
 private:  
 int productId;  
 string productName;  
 float productPrice;  
 public:  
 void get()  
 {  
 cout << "Enter the product ID: ";  
 cin >> productId;  
 cout << "Enter the product name: ";  
 cin >> productName;  
 cout << "Enter the product price: ";  
 cin >> productPrice;  
 }  
 void display()  
 {  
 cout << "Product ID: " << productId << endl;  
 cout << "Product name: " << productName << endl;  
 cout << "Product price: " << productPrice << endl;  
 }  
};  
  
int main()  
{  
 Product obj;  
 obj.get();  
 obj.display();  
 return 0;  
}

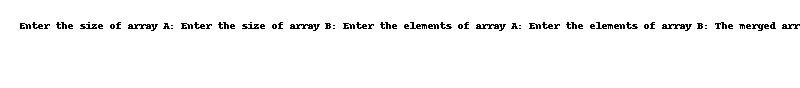
## Output



# Code: 3-7.cpp

/\*  
Suppose A, B, C are arrays of integers of size M, N, and M + N respectively. The numbers in array A  
appear in ascending order while the numbers in array B appear in descending order. Write a user  
defined function in C++ to produce third array C by merging arrays A and B in ascending order. Use A,  
B and C as arguments in the function.  
\*/  
  
#include <iostream>  
using namespace std;  
  
void mergeArrays(int A[], int B[], int C[], int M, int N)  
{  
 int i = 0, j = N - 1, k = 0;  
 while (i < M && j >= 0)  
 {  
 if (A[i] < B[j])  
 {  
 C[k++] = A[i++];  
 }  
 else  
 {  
 C[k++] = B[j--];  
 }  
 }  
 while (i < M)  
 {  
 C[k++] = A[i++];  
 }  
 while (j >= 0)  
 {  
 C[k++] = B[j--];  
 }  
}  
  
int main()  
{  
 int M, N;  
 cout << "Enter the size of array A: ";  
 cin >> M;  
 cout << "Enter the size of array B: ";  
 cin >> N;  
 int A[M], B[N], C[M + N];  
 cout << "Enter the elements of array A: ";  
 for (int i = 0; i < M; i++)  
 {  
 cin >> A[i];  
 }  
 cout << "Enter the elements of array B: ";  
 for (int i = 0; i < N; i++)  
 {  
 cin >> B[i];  
 }  
 mergeArrays(A, B, C, M, N);  
 cout << "The merged array is: ";  
 for (int i = 0; i < M + N; i++)  
 {  
 cout << C[i] << " ";  
 }  
 cout << endl;  
 return 0;  
}

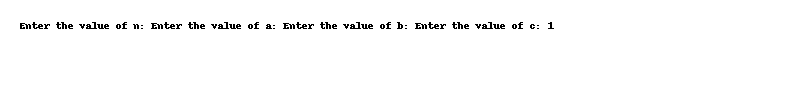
## Output



# Code: 3-8.cpp

/\*  
To celebrate the Reunion of 96 Batch of the Famous School the Ram and Jannu the organizers of the  
event decided to liters of Fruit Drinks. However, an unexpected difficulty occurred in the shop: it  
turned out that Fruit Drinks is sold in bottles 0.5, 1 and 2 li volume. At that, there are exactly 0.5  
bottles in volume, bone-liter bottles and c of two-liter ones. The organizers have enough money to buy  
any amount of Fruit Drinks. What did cause the heated arguments was how many bottles of every kind  
to buy, as this question is pivotal for the of Fruit Drinks among the Friends. Your task is to count the  
number of all the possible ways to buy exactly n liters of Fruit Drinks and persuade the organ this  
number is too large .All the bottles of Fruit Drinks are considered indistinguishable, i.e. two variants of  
buying are different from each other they differ in the number of bottles of at least one kind.  
Constraints:  
1â‰¤nâ‰¤ 10000  
0â‰¤ a, b, c < 5000  
Input Format:  
The first line contains four integers representing, a, b, c respectively. Output Format:  
Print the unique number representing the solution to the problem.  
If it is impossible to buy exactly n liters of Fruit Drinks, print 0.  
\*/  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
 int n, a, b, c;  
 cout << "Enter the value of n: ";  
 cin >> n;  
 cout << "Enter the value of a: ";  
 cin >> a;  
 cout << "Enter the value of b: ";  
 cin >> b;  
 cout << "Enter the value of c: ";  
 cin >> c;  
 int count = 0;  
 for (int i = 0; i <= n; i++)  
 {  
 for (int j = 0; j <= n; j++)  
 {  
 for (int k = 0; k <= n; k++)  
 {  
 if (i \* 0.5 + j + k \* 2 == n && i <= a && j <= b && k <= c)  
 {  
 count++;  
 }  
 }  
 }  
 }  
 cout << count << endl;  
 return 0;  
}

## Output



# Code: 3-9.cpp

/\*  
Tamilnadu Educational Minister has ordered the Director of Higher education to make the Libraries in  
Government schools advanced. So they are planning to create a software which keeps track of the  
books availability and respond to students request for books. Can you help the government to do this?  
Functional Description:  
Input values need to be passed to the Parameterized constructor and to output need to be  
printed by accessing i t.  
Constraints:  
1< roll â‰¤100  
100 â‰¤ bcode< 999  
Input Format:  
First and Second Line of Input has 3 values of type integer, String and Integer separated by a  
space representing  
Roll Number, Name and Book code respectively. Output Format:  
  
Print the Details of Student and Book in the expected format.  
\*/  
  
#include <iostream>  
using namespace std;  
  
class Library  
{  
 private:  
 int roll;  
 string name;  
 int bcode;  
 public:  
 Library(int r, string n, int b)  
 {  
 roll = r;  
 name = n;  
 bcode = b;  
 }  
 void display()  
 {  
 cout << "Roll Number: " << roll << endl;  
 cout << "Name: " << name << endl;  
 cout << "Book Code: " << bcode << endl;  
 }  
};  
  
int main()  
{  
 int roll, bcode;  
 string name;  
 cout << "Enter the roll number, name, and book code: ";  
 cin >> roll >> name >> bcode;  
 Library obj(roll, name, bcode);  
 obj.display();  
 return 0;  
}

## Output

