Q23)

#include <iostream>

#include <cmath>

using namespace std;

class Triangle

{

float a, b, c, base, height;

public:

Triangle()

{

a = 0;

b = 0;

c = 0;

base = 0;

height = 0;

}

Triangle(float x, float y, float z)

{

a = x;

b = y;

c = z;

base = 0;

height = 0;

}

Triangle(float m, float n)

{

base = m;

height = n;

a = 0;

b = 0;

c = 0;

}

float area()

{

if (a == 0 && base == 0)

return 0;

float areaTriangle;

if (a == 0)

areaTriangle = 0.5 \* base \* height;

else

{

float s = (a + b + c) / 2.0;

areaTriangle = sqrt(s \* (s - a) \* (s - b) \* (s - c));

}

return areaTriangle;

}

bool operator==(Triangle T)

{

return a == T.a && b == T.b && c == T.c && base == T.base && height == T.height;

}

void operator=(Triangle &T)

{

a = T.a;

b = T.b;

c = T.c;

base = T.base;

height = T.height;

}

};

int main()

{

Triangle a(4, 3), b(6, 4, 8);

cout << "Area of a : " << a.area() << " sq.units" << endl;

cout << "Area of b : " << b.area() << " sq.units" << endl;

// Assignment operator

Triangle c = b;

cout << "Area of c : " << c.area() << " sq.units" << endl;

// Equality operator

if (c == b)

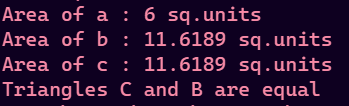
cout << "Triangles C and B are equal" << endl;

else

cout << "Triangles C and B are not equal" << endl;

return 0;

}



Q24)

#include <iostream>

using namespace std;

class Box

{

float length, breadth, height;

public:

friend bool operator==(Box A, Box B);

Box()

{

length = 0;

breadth = 0;

height = 0;

}

Box(float l, float b, float h)

{

length = l;

breadth = b;

height = h;

}

void printBox()

{

cout << "Length : " << length << "units" << endl;

cout << "Length : " << breadth << "units" << endl;

cout << "Height: " << height << "units" << endl;

}

float surfaceArea()

{

return 2 \* (length \* breadth + breadth \* height + height \* breadth);

}

float volume()

{

return length \* breadth \* height;

}

void tellDimension()

{

if (length == breadth && breadth == height)

cout << "Box is cube" << endl;

else

cout << "Box is cuboid" << endl;

}

// Assignment Operator

void operator=(Box &B)

{

length = B.length;

breadth = B.breadth;

height = B.height;

}

// Prefix increment

Box &operator++()

{

length++;

breadth++;

height++;

return \*this;

}

// Postfix increment

Box operator++(int)

{

Box tmp = \*this;

++\*this;

return tmp;

}

// Prefix decrement

Box &operator--()

{

length--;

breadth--;

height--;

return \*this;

}

// Postfix decrement

Box operator--(int)

{

Box tmp = \*this;

--\*this;

return tmp;

}

};

// Equality operator

bool operator==(Box A, Box B)

{

return A.length == B.length && A.breadth == B.breadth && A.height == B.height;

}

int main()

{

float a, b, c;

cout << "Enter sides of box" << endl;

cin >> a >> b >> c;

char choice;

Box box = Box(a, b, c);

while (true)

{

cout << "------MENU------" << endl;

cout << "1.Surface Area" << endl;

cout << "2.Volume" << endl;

cout << "3.Is cuboid/cube" << endl;

cout << "4.Increment box" << endl;

cout << "5.Decrement box" << endl;

cout << "6.Assignment box" << endl;

cout << "7.Check equality of boxes" << endl;

cout << "8.Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

if (choice == '8')

{

cout << "Exiting program..." << endl;

break;

}

else if (choice == '1')

{

cout << box.surfaceArea() << " sq units." << endl;

}

else if (choice == '2')

{

cout << box.volume() << " cu units." << endl;

}

else if (choice == '3')

{

box.tellDimension();

}

else if (choice == '4')

{

cout << "Initially" << endl;

box.printBox();

box++;

cout << "After incrementing" << endl;

box.printBox();

}

else if (choice == '5')

{

cout << "Initially" << endl;

box.printBox();

box--;

cout << "After decrementing" << endl;

box.printBox();

}

else if (choice == '6')

{

Box B;

B = box;

cout << "Dimensions of Box B" << endl;

B.printBox();

}

else if (choice == '7')

{

cout << "Enter dimensions of other box" << endl;

cin >> a >> b >> c;

Box other = Box(a, b, c);

if (other == box)

cout << "Boxes are equal" << endl;

else

cout << "Boxes are not equal" << endl;

}

else

{

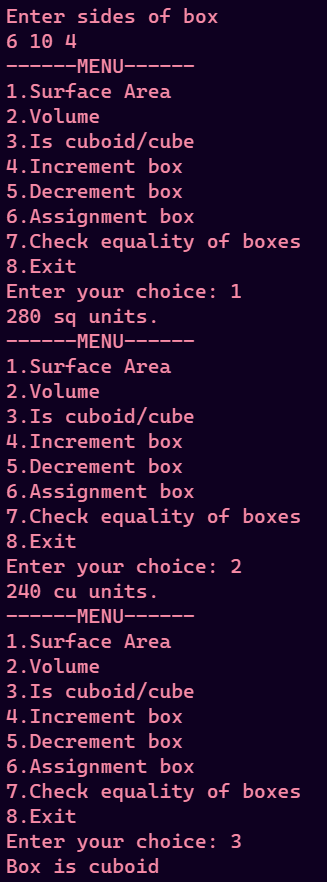
cout << "Enter a valid " << endl;

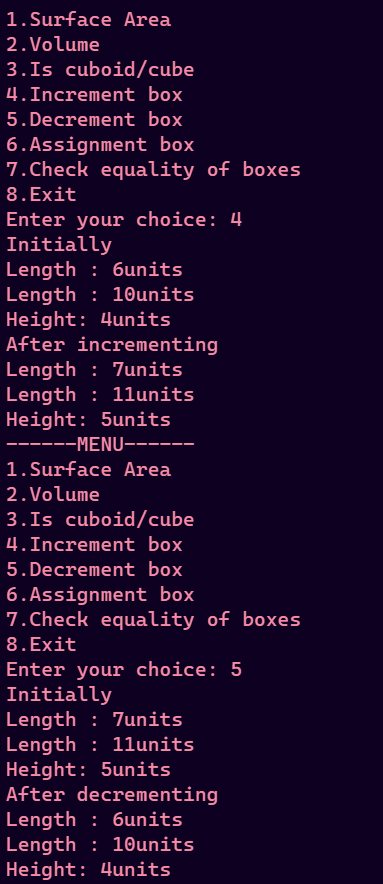
}

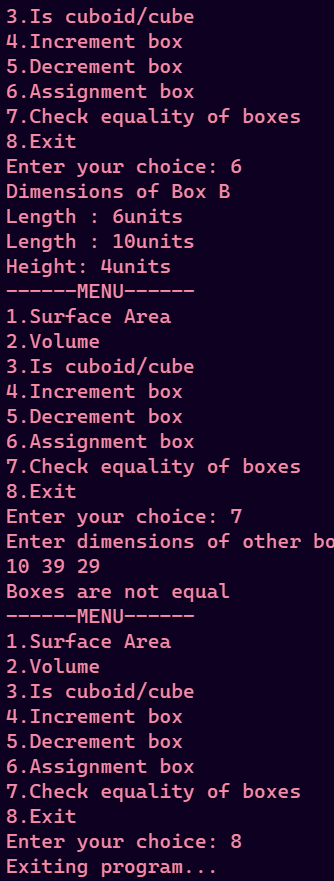
}

return 0;

}







Q28)

#include <iostream>

using namespace std;

void reverseArr(int \*arr, int n)

{

int tmp;

for (int i = 0; i < n / 2; i++)

{

tmp = arr[i];

arr[i] = arr[n - i - 1];

arr[n - i - 1] = tmp;

}

}

int main()

{

int n;

cout << "Enter length of array: ";

cin >> n;

if (n <= 0)

{

cout << "Array should have atleast one element" << endl;

exit(1);

}

int \*a = new int[n];

cout << "Enter array elements" << endl;

for (int i = 0; i < n; i++)

cin >> a[i];

cout << "Initial array: ";

for (int i = 0; i < n; i++)

cout << a[i] << " ";

cout << endl;

reverseArr(a, n);

cout << "Reversed array: ";

for (int i = 0; i < n; i++)

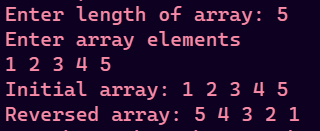
cout << a[i] << " ";

cout << endl;

return 0;

}

Output



Q29)

#include <iostream>

#include <algorithm>

using namespace std;

void printArr(int \*arr, int n)

{

for (int i = 0; i < n; i++)

cout << arr[i] << " ";

cout << endl;

}

void printArrRev(int \*arr, int n)

{

for (int i = n - 1; i >= 0; i--)

cout << arr[i] << " ";

cout << endl;

}

int main()

{

int \*arr = new int[10];

cout << "Enter 10 elements of array" << endl;

for (int i = 0; i < 10; i++)

{

cin >> arr[i];

}

cout << "Orignal Array: ";

printArr(arr, 10);

sort(arr, &arr[10]);

cout << "Ascending Order: ";

printArr(arr, 10);

cout << "Descending order: ";

printArrRev(arr, 10);

return 0;

}

Output

