

Άσκηση 1

Να γραφτεί ένα πρόγραμμα στην C++ στο οποίο να δηλώνεται και να ορίζεται μία συνάρτηση με όνομα `calc_perimeter()`, με τιμή επιστροφής `double`, η οποία θα παίρνει δύο παραμέτρους `length` και `width` τύπου `double` και θα υπολογίζει την περίμετρο ενός παραλληλόγραμμου από την σχέση: ($perimeter = 2 * length + 2 * width$) και μία συνάρτηση `calc_area()` η οποία θα υπολογίζει το εμβαδό από τη σχέση: ($area = length * width$).

Το πρόγραμμα θα ζητά από τον χρήστη να εισάγει τις πλευρές του παραλληλόγραμμου και να επιλέξει την εργασία που θέλει (1->Perimeter, 2->Area), κατόπιν θα καλεί την αντίστοιχη συνάρτηση δίνοντάς της τις τιμές των πλευρών και τέλος θα εκτυπώνει το επιστρεφόμενο αποτελέσματα στην οθόνη. Το πρόγραμμα θα ρωτά τον χρήστη αν θέλει να συνεχίσει τους υπολογισμούς από την αρχή για νέο παραλληλόγραμμο με την προτροπή «Θα βάλεις τιμές για νέο παραλληλόγραμμο? Y(Ναι) , N(Όχι)».

Κώδικας 1

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

double calc_perimeter(double,double);
double calc_area(double,double);

int main()
{
    double length, width, perimeter, area;
    int calculation;
    char selection;
    do
    {
        cout << "\nInsert length: "; cin >> length;
        cout << "Insert width: "; cin >> width;
        cout << "For perimeter press 1, For area press 2: ";
        cin >> calculation;
        if(calculation == 1){
            perimeter = calc_perimeter(length, width);
            cout << "Perimeter is: " << perimeter << endl;
        }
        if(calculation == 2){
            area = calc_area(length, width);
            cout << "Area is: " << area << endl;
        }
        cout << "\nYou will put values for a new rectangle? Y(Yes), N(No): ";
        cin >> selection;
    }while(selection == 'Y' || selection == 'y');

    cout << "End of Program" << endl;

    return 0;
}

double calc_perimeter(double x, double y)
{
    return (2*x) + (2*y);
}

double calc_area(double x, double y)
{
    return x * y;
}
```

Άσκηση 2

Να γραφτεί ένα πρόγραμμα στην C++ το οποίο θα βρίσκει την σχέση μεταξύ 2 ακέραιων αριθμών. Θα χρησιμοποιεί 2 συναρτήσεις *is_multiple()*, *num_relation()* με δύο ορίσματα η καθεμία, χωρίς επιστροφή τιμής (void). Η συνάρτηση *is_multiple* θα ελέγχει εάν ο δεύτερος είναι πολλαπλάσιος του πρώτου και η συνάρτηση *num_relation* θα ελέγχει ποιος από τους δύο είναι μεγαλύτερος (ή ίσοι). Το πρόγραμμα θα ζητά από τον χρήστη να εισάγει τους δύο αριθμούς θα τους στέλνει στις συναρτήσεις ως ορίσματα και αυτές θα εκτυπώνουν τα αποτελέσματα στην οθόνη (δεν θα επιστρέφουν αποτελέσματα).

Κώδικας 2

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

void multiple(int,int);
void relation(int,int);

int main()
{
    int numberOne, numberTwo;
    cout << "Enter first integer: "; cin >> numberOne;
    cout << "Enter second integer: "; cin >> numberTwo;

    multiple(numberOne, numberTwo);
    relation(numberOne, numberTwo);

    cout << endl;
    return 0;
}

void multiple(int _numberOne, int _numberTwo)
{
    if(_numberTwo % _numberOne == 0)
        cout << "The " << _numberTwo << " is a multiple of " << _numberOne << "\n\n";
    else
        cout << "The " << _numberTwo << " is not a multiple of " << _numberOne << "\n\n";
}

void relation(int _numberOne, int _numberTwo)
{
    if (_numberOne > _numberTwo)
        cout<<"The " << _numberOne << " is Biggest"<<endl;
    else if (_numberOne < _numberTwo)
        cout<<"The " << _numberTwo << " is Biggest"<<endl;
    else
        cout<<_numberOne << " and " << _numberTwo << " are equal"<< endl;
}
```