

**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΑΘΛΗΤΙΣΜΟΥ ΚΑΙ ΝΕΟΛΑΙΑΣ  
ΔΙΕΥΘΥΝΣΗ ΜΕΣΗΣ ΤΕΧΝΙΚΗΣ ΚΑΙ ΕΠΑΓΓΕΛΜΑΤΙΚΗΣ ΕΚΠΑΙΔΕΥΣΗΣ  
ΚΑΙ ΚΑΤΑΡΤΙΣΗΣ**

**ΕΝΙΑΙΕΣ ΤΕΛΙΚΕΣ ΠΡΟΑΓΩΓΙΚΕΣ ΚΑΙ  
ΑΠΟΛΥΤΗΡΙΕΣ ΓΡΑΠΤΕΣ ΕΞΕΤΑΣΕΙΣ**

**20 25 - 20 26**

**Γ΄ ΤΑΞΗΣ ΤΕΣΕΚ**

**ΣΕΙΡΑ Α'**

**ΘΕΩΡΗΤΙΚΗ ΚΑΤΕΥΘΥΝΣΗ**

**ΗΜΕΡΟΜΗΝΙΑ : Τετάρτη, 20 Μαΐου 2026**

**ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ: Εφαρμογές Προγραμματισμού III-TEM2**

**ΚΩΔΙΚΟΣ ΜΑΘΗΜΑΤΟΣ : thips302**

**ΛΥΣΕΙΣ**

**ΜΕΡΟΣ Α΄:** Αποτελείται από έξι (6) ερωτήσεις. Κάθε ορθή απάντηση βαθμολογείται με οκτώ (8) μονάδες.

1. Να χαρακτηρίσετε με σωστό (Σ) ή λάθος (Λ) καθεμιά από τις παρακάτω προτάσεις. Όλες οι προτάσεις αφορούν τη γλώσσα προγραμματισμού C++.

A/A	Πρόταση	Σωστό / Λάθος
(α)	Η χειρότερη περίπτωση στη σειριακή αναζήτηση σε ένα πίνακα N στοιχείων απαιτεί N συγκρίσεις.	Σ
(β)	Ο αλγόριθμος ταξινόμησης με εισαγωγή (insertion sort) ταξινομεί τα στοιχεία, χρησιμοποιώντας την αρχή της σύγκρισης και αντιμετάθεσης δύο γειτονικών στοιχείων.	Λ
(γ)	Ο αλγόριθμος ταξινόμησης με εισαγωγή (insertion sort) συγκρίνει πάντα όλα τα στοιχεία μεταξύ τους.	Λ
(δ)	Η δυαδική αναζήτηση έχει καλύτερη απόδοση (απαιτεί λιγότερες συγκρίσεις) από τη σειριακή αναζήτηση σε μεγάλους πίνακες.	Σ

**2 μονάδες για κάθε ορθή απάντηση.**

2. (α) Πώς ονομάζεται η ειδική μέθοδος (συνάρτηση) μέλος που καλείται αυτόματα όταν δημιουργείται ένα αντικείμενο μιας κλάσης; Γράψτε τον αριθμό που αντιστοιχεί στη σωστή απάντηση στον πιο κάτω πίνακα.

- i. Κατασκευαστής (Constructor)
- ii. Καταστροφείας (Destructor)
- iii. Main συνάρτηση
- iv. Μέθοδος (Method)

Απάντηση:	<b>i</b>
-----------	----------

**4 μονάδες η ορθή απάντηση.**

(β) Ποιος είναι ο σκοπός του καταστροφεία (destructor, ~ClassName) στην C++; Γράψτε τον αριθμό που αντιστοιχεί στη σωστή απάντηση στον πιο κάτω πίνακα.

- i. Να αρχικοποιεί μεταβλητές.
- ii. Να απελευθερώνει τους πόρους όταν το αντικείμενο καταστρέφεται.
- iii. Να δημιουργεί αντίγραφα αντικειμένων.
- iv. Να κληρονομεί μεθόδους.

Απάντηση:	<b>ii</b>
-----------	-----------

**4 μονάδες η ορθή απάντηση.**

3. Σας δίνεται ο παρακάτω δισδιάστατος πίνακας ακεραίων αριθμών με όνομα matrix, ο οποίος αποτελείται από τρεις (3) γραμμές και τρεις (3) στήλες:

	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0

Να γράψετε στη γλώσσα προγραμματισμού C++:

(α) Την εντολή που απαιτείται για να δηλωθεί ο πίνακας matrix και να αρχικοποιηθεί με τις συγκεκριμένες τιμές που φαίνονται στον πίνακα του παραπάνω παραδείγματος. Οι απαραίτητες βιβλιοθήκες θεωρούνται ήδη δηλωμένες.

**int matrix[3][3]= {{0, 0, 0},{0, 0, 0},{0, 0, 0}};**

ή

**int matrix[3][3]= {0};**

ή

**int matrix[3][3]= { };**

## **2 Μονάδες**

(β) Τις εντολές που απαιτούνται ώστε να αντικατασταθούν οι αρχικές τιμές του πίνακα matrix με τις συγκεκριμένες τιμές που φαίνονται στον πίνακα του παρακάτω παραδείγματος, χρησιμοποιώντας αποκλειστικά δομή επανάληψης. Οι απαραίτητες βιβλιοθήκες θεωρούνται ήδη δηλωμένες.

	0	1	2
0	2	4	6
1	8	10	12
2	14	16	18

```
int num = 0;
for (int i=0; i<3; i++)
{
    for (int j=0; j<3; j++)
    {
        num = num+2;
        matrix[i][j] = num;
    }
}
```

Η λύση είναι ενδεικτική. Δεκτή οποιαδήποτε λύση χρησιμοποιεί επαναληπτική δομή και οδηγεί στο ορθό γέμισμα των γραμμών του πίνακα.

#### 6 Μονάδες

4. Δίνεται το ακόλουθο τμήμα κώδικα:

```
int N = 10;
int steps = 0;
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        if (i == j) {
            steps++;
        }
    }
}
cout << steps << endl;
```

(α) Πόσες φορές εκτελείται συνολικά η εντολή `steps++` ; Γράψτε τον αριθμό που αντιστοιχεί στη σωστή απάντηση στον πιο κάτω πίνακα.

- i. N
- ii. 2N
- iii.  $N^2$
- iv.  $N^3$

Απάντηση:	i
-----------	---

#### 4 Μονάδες η ορθή απάντηση

(β) Ποια είναι η χρονική πολυπλοκότητα του παραπάνω κώδικα; Γράψτε τον αριθμό που αντιστοιχεί στη σωστή απάντηση στον πιο κάτω πίνακα.

- i.  $O(1)$
- ii.  $O(N)$
- iii.  $O(N^2)$
- iv.  $O(N^3)$

Απάντηση:
-----------

iii
-----

#### 4 Μονάδες η ορθή απάντηση

5. Τι θα εμφανίσει στην οθόνη το πιο κάτω πρόγραμμα;

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

class A {
    int x = 10;
public:
    int z = 20;

    A(int a) {
        x = a;
    }

    int inc( ) {
        x++;
        return x;
    }

    void test( ) {
        cout << "X=" << x << endl;
    }

    int test2( ) {
        return z + 1;
    }
};

int main( ) {
    A obj(5);
    obj.test( );
    obj.z = 5;
    cout << obj.inc( ) << endl;
    cout << obj.inc( ) << endl;
    cout << obj.test2( );
    return 0;
}
```

**X=5**

**6**

**7**

**6**

**2 Μονάδες κάθε ορθή γραμμή έξοδου (σύνολο 8 μονάδες).**

6. Δίνεται το ακόλουθο τμήμα κώδικα:

```
ifstream file("data.txt");
```

```
int x;
```

```
file >> x;
```

```
cout << x;
```

Τι θα συμβεί αν το αρχείο (data.txt) δεν υπάρχει; Γράψτε το γράμμα που αντιστοιχεί στην σωστή απάντηση στον πιο κάτω πίνακα.

- (α) Το πρόγραμμα θα τερματιστεί επιστρέφοντας συντακτικό σφάλμα.
- (β) Το πρόγραμμα θα εκτελεστεί και θα δημιουργηθεί ένα νέο, άδειο αρχείο.
- (γ) Το πρόγραμμα θα παρουσιάσει σφάλμα εκτέλεσης.
- (δ) Το πρόγραμμα θα εκτελεστεί και θα τυπωθεί στην οθόνη ως έξοδος μια τυχαία τιμή.

Απάντηση:	(δ)
-----------	-----

**8 Μονάδες η ορθή απάντηση**

**ΜΕΡΟΣ Β΄**: Αποτελείται από τέσσερις (4) ερωτήσεις. Κάθε ορθή απάντηση βαθμολογείται με δέκα (10) μονάδες.

7. Να δημιουργήσετε πρόγραμμα στη γλώσσα προγραμματισμού C++, το οποίο:

(α) Να καταχωρίζει τα ονόματα πέντε (5) υποψηφίων σε ένα μονοδιάστατο πίνακα με το όνομα `ipropsifios` και τον αριθμό των σταυρών προτίμησης που αντιστοιχεί σε κάθε υποψήφιο σε ένα άλλο παράλληλο μονοδιάστατο πίνακα με το όνομα `psifoi`. Στα ονόματα των υποψηφίων δεν επιτρέπονται τα κενά. Αντί κενού να χρησιμοποιείται το σύμβολο `_`.

(β) Να χρησιμοποιεί τη συνάρτηση `katataxi_ipropsifion`, η οποία θα λαμβάνει ως παραμέτρους από το κύριο πρόγραμμα (`main`) τους πίνακες `ipropsifios` και `psifoi` και θα τους ταξινομεί σε φθίνουσα σειρά με βάση τον αριθμό των ψήφων με τη μέθοδο της φυσαλίδας (`bubble sort`).

(γ) Να χρησιμοποιεί τη συνάρτηση `print_results`, η οποία θα τυπώνει τους ταξινομημένους πίνακες στην οθόνη με δεξιά στοίχιση 30 χαρακτήρων, όπως φαίνεται στο πιο κάτω παράδειγμα.

Παράδειγμα Εξόδου:

```
----- ΑΠΟΤΕΛΕΣΜΑΤΑ -----
                Υποψήφιος                Ψήφοι
Υποψήφιος_4                134
Υποψήφιος_3                78
Υποψήφιος_5                30
Υποψήφιος_1                23
Υποψήφιος_2                11
```

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
#define N 5
```

```
void katataxi_ipropsifion(string arrNames[ ], int arrVotes[ ]);
```

```
void print_results(string arrNames[ ], int arrVotes[ ]);
```

```
int main() {
```

```
    string ipropsifios[N];
```

```
    int psifoi[N];
```

```
for (int i = 0; i < N; i++) {
```

```
    cout << "Δώσε Όνομα Υποψηφίου (χωρίς κενά, χρησιμοποιήστε _): ";
```

```
    cin >> ipropsifios[i];
```

```

        cout << "Δώσε τον αριθμό έγκυρων ψήφων του Υποψηφίου: ";
        cin >> psifoi[i];
        cout << endl;
    }
    katataxi_ipopsifion(ipopsifios, psifoi);
    print_results(ipopsifios, psifoi);
    return 0;
}

void katataxi_ipopsifion(string arrNames[], int arrVotes[]) {
    bool sorted;
    do {
        sorted = true;
        for (int i = 0; i < N - 1; i++) {
            if (arrVotes[i] < arrVotes[i + 1]) {
                int tempVotes = arrVotes[i];
                arrVotes[i] = arrVotes[i + 1];
                arrVotes[i + 1] = tempVotes;
                string tempName = arrNames[i];
                arrNames[i] = arrNames[i + 1];
                arrNames[i + 1] = tempName;
                sorted = false;
            }
        }
    } while (sorted==false);
}

void print_results(string arrNames[ ], int arrVotes[ ]) {
    cout << "----- ΑΠΟΤΕΛΕΣΜΑΤΑ -----" << endl;
    cout << setw(30) << "Υποψήφιος"
        << setw(30) << "Ψήφοι" << endl;
    for (int i = 0; i < N; i++) {
        cout << setw(30) << arrNames[i]
            << setw(30) << arrVotes[i] << endl;
    }
}

```

Η λύση είναι ενδεικτική. Οποιαδήποτε άλλη επιστημονικά τεκμηριωμένη λύση είναι αποδεκτή.

Η πιο πάνω λύση συμπεριλαμβάνει λεπτομερή μηνύματα προς το χρήστη για την είσοδο δεδομένων τα οποία δεν αναμένονται στις λύσεις των μαθητών.

**(α) 3 ΜΟΝΑΔΕΣ**

- 1 Μονάδα οι ορθές δηλώσεις μεταβλητών
- 2 Μονάδες η/οι φωλιασμένη/ες επανάληψη/εις (nested for loop(s)) για την είσοδο δεδομένων. Τα ολοκληρωμένα μηνύματα προς τον χρήστη δεν αναμένονται στις απαντήσεις των μαθητών λαμβάνοντας υπόψη ότι ο χρόνος είναι περιορισμένος και η λύση δίνεται στο χαρτί.

**(β) 4 ΜΟΝΑΔΕΣ**

- 1 Μονάδα ο ορθός ορισμός της συνάρτησης συμπεριλαμβανομένου και των παραμέτρων. Τα μεγέθη για τους πίνακες στις παραμέτρους δεν είναι απαραίτητα.
- 0.5 Μονάδα η ορθή επανάληψη for
- 0.5 Μονάδα η ορθή συνθήκη για την αύξουσα σειρά ταξινόμησης
- 1 Μονάδα η ορθή αντιμετάθεση ψήφων
- 1 Μονάδα η ορθή αντιμετάθεση ονομάτων

**(γ) 3 ΜΟΝΑΔΕΣ**

- 1 Μονάδα ο ορθός ορισμός της συνάρτησης συμπεριλαμβανομένου και των παραμέτρων. Τα μεγέθη για τους πίνακες στις παραμέτρους δεν είναι απαραίτητα.
- 2 Μονάδες η ορθή εκτύπωση των αποτελεσμάτων
  - 0.5 Μονάδα η ορθή παρουσίαση των τίτλων
  - 1 Μονάδα η ορθή παρουσίαση των αποτελεσμάτων
  - 0.5 Μονάδα η ορθή στοίχιση των αποτελεσμάτων

**Αφαιρετικά γενικά**

-0.25 αν λείπει το `#include<iostream>`

-0.25 αν λείπει το `#include<iomanip>`

-0.25 αν λείπει το `using namespace std;`

-0.25 αν λείπει το `int main()`

-0.25 αν λείπει { ή και }

-0.25 αν λείπουν δυο ή περισσότερα ; (μια φορά)

8. Πιο κάτω παρουσιάζονται δύο συναρτήσεις στη γλώσσα προγραμματισμού C++. Η πρώτη συνάρτηση εκτελεί ταξινόμηση μονοδιάστατου πίνακα ακέραιων αριθμών κατά αύξουσα σειρά χρησιμοποιώντας τον αλγόριθμο ταξινόμησης με εισαγωγή (insertion sort). Η δεύτερη εκτελεί αναζήτηση ενός ακέραιου αριθμού στον ταξινομημένο πίνακα χρησιμοποιώντας τον αλγόριθμο δυαδικής αναζήτησης (binary search). Αν εντοπιστεί ο αριθμός, επιστρέφει την τιμή true, ενώ αν δεν εντοπιστεί ο αριθμός επιστρέφει την τιμή false. Θεωρείστε ότι στον πίνακα δεν υπάρχουν επαναλαμβανόμενες τιμές. Στις συναρτήσεις υπάρχουν τέσσερα (4) λογικά λάθη. Να τα εντοπίσετε και να τα διορθώσετε συμπληρώνοντας στον παρακάτω πίνακα τον αριθμό της γραμμής και τη διορθωμένη εντολή. Στις συναρτήσεις να μη γίνει καμία προσθήκη ή αφαίρεση εντολής.

```
1. void insertionSort(int arr[ ], int N) {
2.     int temp;
3.     int j;
4.     for (int i=1; i<=N; i++) {
5.         temp = arr[i];
6.         j = i - 1;
7.         while(j>0 && arr[j]>temp) {
8.             arr[j+1] = arr[j];
9.             j--;
10.        }
11.        arr[j] = temp;
12.    }
13. }
14. bool binary_search(int arr[ ], int N, int target) {
15.     int first = 0, last = N-1, mid;
16.     while (first <= last) {
17.         mid = (first + last) / 2;
18.         if (target == arr[mid])
19.             return true;
20.         else if (target < arr[mid])
21.             last = mid + 1;
22.         else
23.             first = mid + 1;
24.     }
25.     return false;
26. }
```

A/A	Αριθμός Γραμμής	Διορθωμένη εντολή
1	4	for (int i=1; i<N; i++) {
2	7	while(j>=0 && arr[j]>temp){
3	11	arr[j+1] = temp;
4	21	last = mid - 1;

**1.25 Μονάδες για τον εντοπισμό της γραμμής του κάθε λάθους και 1.25 Μονάδες για κάθε ορθή διόρθωση του λάθους (σύνολο 10 Μονάδες)**

9. Δίνεται ο παρακάτω κώδικας στη γλώσσα προγραμματισμού C++. Ο κώδικας επιχειρεί να υλοποιήσει τον αλγόριθμο ταξινόμησης με τη μέθοδο της φυσαλίδας (*bubble sort*) στη βέλτιστη μορφή του, για την ταξινόμηση ενός μονοδιάστατου πίνακα ακεραίων αριθμών σε αύξουσα σειρά.

```
void bubblesort(int arr[ ], int N) {
    bool sorted;
    int temp, S = N;
    do {
        sorted = true;
        for ( int i=0; i<S-1; i++) {
            if (  ) {
                temp = arr[i];
                arr[i] = arr[i+1];
                
                sorted = false;
            }
        }
        
    } while (  );
}
```

Να γράψετε:

(α) Τη συνθήκη που πρέπει να τοποθετηθεί στη θέση A, ώστε ο αλγόριθμος να ταξινομεί τον πίνακα σε αύξουσα σειρά.

**arr[i] > arr[i+1]**

(β) Την εντολή που πρέπει να τοποθετηθεί στη θέση B, ώστε να πραγματοποιείται σωστά η αντιμετάθεση των στοιχείων με τη χρήση της βοηθητικής μεταβλητής temp.

**arr[i+1] = temp;**

(γ) Την εντολή που πρέπει να τοποθετηθεί στη θέση Γ, ώστε να μειώνεται το μέγεθος του τμήματος του πίνακα που πρέπει να εξεταστεί μετά από κάθε πέρασμα.

**S--;**

(δ) Τη συνθήκη που πρέπει να τοποθετηθεί στη θέση Δ, ώστε ο βρόχος να επαναλαμβάνεται μόνο όταν απαιτείται.

**sorted == false**

## 2.5 Μονάδες για κάθε ορθή απάντηση

10. Δίνεται το πιο κάτω πρόγραμμα σε γλώσσα προγραμματισμού C++, το οποίο υλοποιεί μια κλάση με όνομα Sum. Η κλάση Sum χρησιμοποιείται για την αποθήκευση και επεξεργασία τριών ακέραιων τιμών.

Στο πρόγραμμα υπάρχουν κενά, τα οποία πρέπει να συμπληρωθούν με βάση τα πιο κάτω ζητούμενα. Για κάθε κενό σημείο συμπληρώστε τον σωστό κώδικα στον πίνακα απαντήσεων.

```
#include <iostream>
using namespace std;
class Sum {
private:
    int x, y;
    A :
    int z;
    B (int a, int b) {
x = a;
y = b;
}
    Γ add() {
if (x + y > z)
    return x + y + z;
else
    Δ ;
}
    E display() {
cout << "Result:" << endl;
}
};
int main() {
    Z s( H );
    Θ ;
    I ;
    cout << K ;
    return 0;
}
```

Πίνακας Απαντήσεων

Κενό	Σωστός κώδικας
<b>A</b>	<b>public</b>
<b>B</b>	<b>Sum</b>
<b>Γ</b>	<b>int</b>
<b>Δ</b>	<b>return -1</b>
<b>E</b>	<b>void</b>
<b>Z</b>	<b>Sum</b>
<b>H</b>	<b>8 , 7</b>
<b>Θ</b>	<b>s.z = 20</b>
<b>I</b>	<b>s.display()</b>
<b>K</b>	<b>s.add()</b>

**1 Μονάδα κάθε ορθή απάντηση**

Ζητούμενα:

(α) Να γράψετε το προσδιοριστικό πρόσβασης, ώστε η μεταβλητή  $z$  να είναι προσβάσιμη από το κύριο πρόγραμμα `main()`.

(β) Να συμπληρώσετε την εντολή για την δημιουργία του κατασκευαστή της κλάσης `Sum` ο οποίος δέχεται δύο ακέραιες τιμές  $a$  και  $b$  και αρχικοποιεί τις μεταβλητές  $x$  και  $y$ .

(γ) Να συμπληρώσετε τον τύπο επιστροφής και την εντολή στο τμήμα `else` της μεθόδου `add()` έτσι ώστε να ελέγχει αν το άθροισμα των μεταβλητών  $x$  και  $y$  είναι μεγαλύτερο από την τιμή της  $z$ , να επιστρέφει το άθροισμα  $x + y + z$ , διαφορετικά να επιστρέφει την τιμή  $-1$ .

(δ) Να συμπληρώσετε τις εντολές για την υλοποίηση της μεθόδου `display()` η οποία εμφανίζει στην οθόνη το μήνυμα "Result:".

(ε) Να συμπληρώσετε τις εντολές στο κύριο πρόγραμμα `main()` για:

- i. να δημιουργηθεί το αντικείμενο  $s$  της κλάσης `Sum` με τιμές  $a = 8$  και  $b = 7$ ,
- ii. να αποδοθεί στη μεταβλητή  $z$  η τιμή  $20$  η οποία πρέπει να πάρει τιμή πριν χρησιμοποιηθεί στον έλεγχο της μεθόδου `add()`,
- iii. να κληθεί η μέθοδος `display()`,
- iv. να εμφανιστεί στην οθόνη το αποτέλεσμα της μεθόδου `add()`.

**Τελική έξοδος του προγράμματος (Οθόνη):**

Result: -1
---------------

**ΜΕΡΟΣ Γ΄: Αποτελείται από μία (1) ερώτηση. Η ορθή απάντηση βαθμολογείται με δώδεκα (12) μονάδες.**

11. Στον 1ο Παγκύπριο Διαγωνισμό Τεχνικών Δεξιοτήτων του κλάδου Ηλεκτρολογίας και Ηλεκτρονικών Εφαρμογών στη διαγωνιστική κατηγορία ρομποτικής, λαμβάνουν μέρος 15 ομάδες από όλες τις τεχνικές σχολές της Κύπρου. Για τη διαχείριση των δεδομένων των ομάδων, χρησιμοποιώντας αντικειμενοστραφή προγραμματισμό, να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού C++ το οποίο:

(α) Να δημιουργεί μια κλάση με όνομα Team, η οποία να αναπαριστά μία ομάδα.

(1 Μονάδα)

(β) Στα ιδιωτικά μέλη της κλάσης να περιλαμβάνονται:

- i. Πίνακας βαθμολογιών για τρεις (3) προκλήσεις (ακέραιος αριθμός, 0-100): Scores[3]
- ii. Πίνακας παραβάσεων (ακέραιος αριθμός 1 στην περίπτωση μη παράβασης ή ακέραιος αριθμός 0 στην περίπτωση παράβασης) για κάθε πρόκληση: Violation[3]
- iii. Συνολικός χρόνος ολοκλήρωσης (πραγματικός αριθμός): Time

(1.5 Μονάδες)

(γ) Στα δημόσια μέλη της κλάσης να περιλαμβάνονται:

- i. Όνομα ομάδας: Name
- ii. Τελική βαθμολογία (πραγματικός αριθμός): Total
- iii. Κατασκευαστής (Constructor) ο οποίος να αρχικοποιεί:
  - Όλες τις βαθμολογίες σε 0.
  - Όλες τις παραβάσεις σε 1.
  - Τον χρόνο και την τελική βαθμολογία σε 0.
- iv. Μέθοδος ReadData() η οποία να:
  - Εισάγει το όνομα της ομάδας.
  - Εισάγει τις βαθμολογίες για τις 3 προκλήσεις.
  - Εισάγει τις παραβάσεις (0 ή 1).
  - Εισάγει τον συνολικό χρόνο ολοκλήρωσης.
- v. Μέθοδος CalculateTotalScore() η οποία να:
  - Μηδενίζει τη βαθμολογία πρόκλησης αν υπάρχει παράβαση.
  - Υπολογίζει τον μέσο όρο των 3 βαθμολογιών.
  - Διαιρεί τον μέσο όρο με τον συνολικό χρόνο.
  - Αποθηκεύει το αποτέλεσμα στο Total.

(3.5 Μονάδες)

(δ) Να δημιουργηθεί εκτός κλάσης, κατάλληλη συνάρτηση ShowResults(), η οποία να ταξινομεί τις ομάδες κατά φθίνουσα σειρά σύμφωνα με την τελική βαθμολογία και στη συνέχεια να εμφανίζει τα ονόματα των ομάδων παράλληλα με τις τελικές βαθμολογίες.

(3 Μονάδες)

(ε) Στο κύριο πρόγραμμα main( ):

- i. Να δημιουργηθεί πίνακας 15 αντικειμένων τύπου Team.
- ii. Να καλείται η μέθοδος ReadData() για κάθε ομάδα (με επανάληψη, π.χ. μέσα σε for).
- iii. Να καλείται η μέθοδος CalculateTotalScore() για κάθε ομάδα (με επανάληψη, π.χ. μέσα σε for).
- iv. Να καλείται η μέθοδος ShowResults() για ταξινόμηση και εμφάνιση των αποτελεσμάτων όπως διατυπώνεται στο ερώτημα (δ).

(3 Μονάδες)

<u>Παράδειγμα εισόδου (για 3 ομάδες)</u>	<u>Παράδειγμα εξόδου</u>
<p>ΟΜΑΔΑ 1 Δώσε όνομα ομάδας: ΤΕΣΕΚ01 Βαθμολογία πρόκλησης 1: 80 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 1 Βαθμολογία πρόκλησης 2: 75 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 0 Βαθμολογία πρόκλησης 3: 85 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 1 Συνολικός χρόνος: 66</p> <p>ΟΜΑΔΑ 2 Δώσε όνομα ομάδας: ΤΕΣΕΚ02 Βαθμολογία πρόκλησης 1: 90 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 1 Βαθμολογία πρόκλησης 2: 93 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 1 Βαθμολογία πρόκλησης 3: 78 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 1 Συνολικός χρόνος: 55</p> <p>ΟΜΑΔΑ 3 Δώσε όνομα ομάδας: ΤΕΣΕΚ03 Βαθμολογία πρόκλησης 1: 91 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 1 Βαθμολογία πρόκλησης 2: 89 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 1 Βαθμολογία πρόκλησης 3: 73 Παράβαση (1=ΟΧΙ, 0=ΝΑΙ): 1 Συνολικός χρόνος: 56</p>	<p>--- ΑΠΟΤΕΛΕΣΜΑΤΑ --- 1. ΤΕΣΕΚ02 - Βαθμολογία: 1.58182 2. ΤΕΣΕΚ03 - Βαθμολογία: 1.50595 3. ΤΕΣΕΚ01 - Βαθμολογία: 0.833333</p>

```

#include <iostream>
using namespace std;
#define pr 3

class Team {
private:

    int Scores[pr];
    int Violation[pr];
    float Time;

public:
    string Name;
    float Total;
    // Constructor
    Team() {
        Time = 0;
        Total = 0;
        for (int i = 0; i < pr; i++) {
            Scores[i] = 0;
            Violation[i] = 1;
        }
    }

    void ReadData() {
        cout << "Δώσε όνομα ομάδας: ";
        cin >> Name;

        for (int i = 0; i < pr; i++) {
            cout << "Βαθμολογία πρόκλησης " << i+1 << ": ";
            cin >> Scores[i];

            cout << "Παράβαση (1=OXI, 0=NAI): ";
            cin >> Violation[i];
        }

        cout << "Συνολικός χρόνος: ";
        cin >> Time;
    }

    void CalculateTotalScore() {
        float sum = 0;

        for (int i = 0; i < pr; i++) {
            if (Violation[i] == 0) {
                Scores[i] = 0;
            }
        }
    }
};

```

```

        }
        sum += Scores[i];
    }

    float average = sum / pr;

    if (Time != 0)
        Total = average / Time;
    else
        Total = 0;
}

};

void ShowResults(Team teams[ ], int n) {
    // Bubble sort
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-1-i; j++) {
            if (teams[j].Total < teams[j+1].Total) {
                Team temp = teams[j];
                teams[j] = teams[j+1];
                teams[j+1] = temp;
            }
        }
    }

    cout << endl << "--- ΑΠΟΤΕΛΕΣΜΑΤΑ ---" << endl;
    for (int i = 0; i < n; i++) {
        cout << i+1 << ". "
            << teams[i].Name << " - Βαθμολογία: " << teams[i].Total << endl;
    }
}

int main() {
    const int N = 3;
    Team teams[N];

    for (int i = 0; i < N; i++) {
        cout << endl << "ΟΜΑΔΑ " << i+1 << endl;
        teams[i].ReadData();
    }

    for (int i = 0; i < N; i++) {
        teams[i].CalculateTotalScore();
    }
}

```

```
ShowResults(teams, N);  
  
return 0;  
}
```

Η λύση είναι ενδεικτική. Οποιαδήποτε άλλη επιστημονικά τεκμηριωμένη λύση είναι αποδεκτή.

(α) 1 Μονάδα η ορθή δημιουργία της κλάσης συμπεριλαμβανομένου του τερματισμού της ( }; ).

(β) 0.5 Μονάδα ο ορθός ορισμός κάθε μεταβλητής (σύνολο 1.5 μονάδες).

(γ) 0.25 της Μονάδας ο ορθός ορισμός κάθε μεταβλητής, 1 Μονάδα η ορθή υλοποίηση της κάθε μεθόδου (σύνολο 3.5 μονάδες).

(δ) 1.5 Μονάδες η ορθή σύνταξη της μεθόδου ταξινόμησης και 1.5 Μονάδα η ορθή σύνταξη της μεθόδου παρουσίασης (σύνολο 3 μονάδες).

(ε) 0.5 Μονάδα ο ορθός ορισμός των μεταβλητών, 1 μονάδα η επαναληπτική δομή και η σωστή κλήση της μεθόδου ReadData(), 1 μονάδα η επαναληπτική δομή και η σωστή κλήση της μεθόδου CalculateTotalScore(), 0.5 Μονάδα η σωστή κλήση της μεθόδου ShowResults(teams, N) (σύνολο 3 μονάδες).

**Αφαιρετικά (γενικά)**

- 0.25 αν λείπει το `#include <iostream>`
- 0.25 αν λείπει το `using namespace std;`
- 0.25 αν λείπει το `int main ( )`
- 0.25 αν λείπει { ή και }
- 0.25 αν λείπουν δύο ή περισσότερα ; (μια φορά)