

**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**

**ΣΥΝΟΛΙΚΗ ΔΙΑΡΚΕΙΑ ΓΡΑΠΤΗΣ ΕΞΕΤΑΣΗΣ: 90΄ λεπτά**

**ΤΟ ΕΞΕΤΑΣΤΙΚΟ ΔΟΚΙΜΙΟ ΑΠΟΤΕΛΕΙΤΑΙ ΑΠΟ ΔΕΚΑΕΠΤΑ ( 17 ) ΣΕΛΙΔΕΣ.**

**ΤΑ ΜΕΡΗ ΤΟΥ ΕΞΕΤΑΣΤΙΚΟΥ ΔΟΚΙΜΙΟΥ ΕΙΝΑΙ ΤΡΙΑ (Α΄, Β΄ ΚΑΙ Γ΄).**

**ΟΔΗΓΙΕΣ (για τους εξεταζόμενους)**

- 1. Να απαντήσετε ΟΛΑ τα ερωτήματα πάνω στο εξεταστικό δοκίμιο.**
- 2. Να μη γράψετε πουθενά το όνομα σας στο εξεταστικό δοκίμιο εκτός του καθορισμένου χώρου στο χαρτονάκι που σας έχει δοθεί.**
- 3. Να απαντήσετε σε όλα τα θέματα μόνο με πένα χρώματος μπλε ανεξίτηλης μελάνης. Μολύβι επιτρέπεται, μόνο αν το ζητάει η εκφώνηση, και μόνο για σχήματα, πίνακες, διαγράμματα κλπ.**
- 4. Απαγορεύεται η χρήση διορθωτικού υγρού και διορθωτικής ταινίας.**
- 5. Επιτρέπεται η χρήση μη προγραμματιζόμενης υπολογιστικής μηχανής.**

**ΣΑΣ ΕΥΧΟΜΑΣΤΕ ΚΑΛΗ ΕΠΙΤΥΧΙΑ**

**ΟΔΗΓΙΕΣ (για την επιτροπή εξετάσεων)**

- 1. Το εξεταστικό δοκίμιο να εκτυπωθεί και στις δύο όψεις.**

**ΤΟ ΕΞΕΤΑΣΤΙΚΟ ΔΟΚΙΜΙΟ ΝΑ ΕΚΤΥΠΩΘΕΙ: **ΜΑΥΡΟΑΣΠΡΟ****

**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.	
(β)	Ο αλγόριθμος ταξινόμησης με εισαγωγή (insertion sort) ταξινομεί τα στοιχεία, χρησιμοποιώντας την αρχή της σύγκρισης και αντιμετάθεσης δύο γειτονικών στοιχείων.	
(γ)	Ο αλγόριθμος ταξινόμησης με εισαγωγή (insertion sort) συγκρίνει πάντα όλα τα στοιχεία μεταξύ τους.	
(δ)	Η δυαδική αναζήτηση έχει καλύτερη απόδοση (απαιτεί λιγότερες συγκρίσεις) από τη σειριακή αναζήτηση σε μεγάλους πίνακες.	

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. Καταστροφέας (Destructor)
- iii. Main συνάρτηση
- iv. Μέθοδος (Method)

Απάντηση:	
-----------	--

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

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

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
matrix	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.

---

---

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

---

---

---

---

---

---

---

---

---

---

(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:	
-----------	--

(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:	
-----------	--

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;
}
```

---

---

---

---

---

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:	
-----------	--







```

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		
2		
3		
4		

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 Α, ώστε ο αλγόριθμος να ταξινομεί τον πίνακα σε αύξουσα σειρά.

---

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

---

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

---

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>B</b>	
<b>Γ</b>	
<b>Δ</b>	
<b>E</b>	
<b>Z</b>	
<b>H</b>	
<b>Θ</b>	
<b>I</b>	
<b>K</b>	

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 ---</p> <ol style="list-style-type: none"><li>1. ΤΕΣΕΚ02 - Βαθμολογία: 1.58182</li><li>2. ΤΕΣΕΚ03 - Βαθμολογία: 1.50595</li><li>3. ΤΕΣΕΚ01 - Βαθμολογία: 0.833333</li></ol>





