

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΑΘΛΗΤΙΣΜΟΥ
ΚΑΙ ΝΕΟΛΑΙΑΣ
ΔΙΕΥΘΥΝΣΗ ΑΝΩΤΕΡΗΣ ΕΚΠΑΙΔΕΥΣΗΣ
ΥΠΗΡΕΣΙΑ ΕΞΕΤΑΣΕΩΝ

ΠΑΓΚΥΠΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΠΡΟΣΒΑΣΗΣ 2026

Μάθημα: ΠΛΗΡΟΦΟΡΙΚΗ (15)

Ημερομηνία και ώρα εξέτασης: Τετάρτη, 17 Ιουνίου 2026

08:00 – 11:00

ΤΟ ΕΞΕΤΑΣΤΙΚΟ ΔΟΚΙΜΙΟ ΑΠΟΤΕΛΕΙΤΑΙ ΑΠΟ ΔΕΚΑΠΕΝΤΕ (15) ΣΕΛΙΔΕΣ

ΟΔΗΓΙΕΣ

- Να απαντήσετε σε **όλες** τις ερωτήσεις.
- Το εξεταστικό δοκίμιο αποτελείται από τρία (3) μέρη: **A'**, **B'** και **Γ'**.
- Το **μέρος A'** αποτελείται από **έξι (6)** ασκήσεις και κάθε άσκηση βαθμολογείται με **πέντε (5)** μονάδες.
- Το **μέρος B'** αποτελείται από **τέσσερις (4)** ασκήσεις και κάθε άσκηση βαθμολογείται με **δέκα (10)** μονάδες.
- Το **μέρος Γ'** αποτελείται από **δύο (2)** ασκήσεις και κάθε άσκηση βαθμολογείται με **δεκαπέντε (15)** μονάδες.
- Επισυνάπτεται **τυπολόγιο** που αποτελείται από μια (1) σελίδα.
- Επιτρέπεται η χρήση **μη προγραμματιζόμενης** υπολογιστικής μηχανής.
- Οι μοναδικές βιβλιοθήκες που επιτρέπονται στη δημιουργία προγραμμάτων, είναι οι **<iostream>**, **<fstream>**, **<string>**, **<iomanip>**, **<cmath>** και **<climits>**.
- Η έκδοση της γλώσσας προγραμματισμού C++ που μπορεί να χρησιμοποιήσει ο υποψήφιος είναι η **C++98 (ISO/IEC 14882:1998)**. Οποιοσδήποτε **επεκτάσεις (extensions)** παρέχονται από κάποιους μεταγλωττιστές (compilers) **δεν** μπορούν να χρησιμοποιηθούν.

Τα σύμβολα των λογικών κυκλωμάτων, καθώς και το λεκτικό περιεχόμενό τους μπορούν να γίνουν με μολύβι.

ΜΕΡΟΣ Α´

ΑΣΚΗΣΗ 1:

Δίνεται το πιο κάτω πρόγραμμα στη γλώσσα προγραμματισμού C++:

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main() {
int n1, n2, n3, n4;
float r1, r2;
cin >> n1 >> n2 >> n3;
```

A

```
if( B ) {
```

Γ

Δ

```
}
```

E

```
return 0;
}
```

Να γράψετε στο τετράδιο απαντήσεών σας:

(α) Την εντολή που πρέπει να τοποθετηθεί στη **θέση A**, έτσι ώστε να εκχωρείται στην μεταβλητή **r1** ο μέσος όρος των τιμών των μεταβλητών **n1**, **n2** και **n3**.

(Μονάδα 1)

(β) Τη συνθήκη που πρέπει να τοποθετηθεί στη **θέση B**, η οποία ελέγχει αν η μεταβλητή **n1** είναι άνιση με την μεταβλητή **n2** ή αν η μεταβλητή **n1** είναι ίση με την μεταβλητή **n3**.

(Μονάδα 1)

(γ) Την εντολή που πρέπει να τοποθετηθεί στη **θέση Γ**, έτσι ώστε να εκχωρείται στην μεταβλητή **n4** το άθροισμα των ψηφίων των μονάδων των μεταβλητών **n1** και **n2**.

(Μονάδα 1)

(δ) Την εντολή που πρέπει να τοποθετηθεί στη **θέση Δ**, έτσι ώστε να εκχωρείται στην μεταβλητή **r2** το ακέραιο μέρος της μεταβλητής **r1**.

(Μονάδα 1)

(ε) Την εντολή που πρέπει να τοποθετηθεί στη **θέση E**, έτσι ώστε να τυπώνεται η τιμή της μεταβλητής **r1** με ακρίβεια δύο (2) δεκαδικών ψηφίων.

(Μονάδα 1)

ΑΣΚΗΣΗ 2:

Δίνονται στο **δεκαδικό** σύστημα αρίθμησης ο πραγματικός αριθμός **A=38.25** και στο **δυναδικό** σύστημα αρίθμησης οι αριθμοί **B=10001010** και **Γ=00101000**.

(α) Να δείξετε ότι η αντίστοιχη τιμή του **πραγματικού αριθμού A** στο δυναδικό σύστημα αρίθμησης είναι **(100110.01)₂**, σημειώνοντας τα βήματα που ακολουθήσατε για να φτάσετε στο συγκεκριμένο αποτέλεσμα.

(Μονάδες 2)

(β) Να δείξετε ότι η αντίστοιχη τιμή του **δυναδικού αριθμού B** στο δεκαδικό σύστημα αρίθμησης είναι **(138)₁₀**, σημειώνοντας τα βήματα που ακολουθήσατε για να φτάσετε στο συγκεκριμένο αποτέλεσμα.

(Μονάδα 1)

(γ) Χρησιμοποιώντας το **συμπλήρωμα ως προς 2** του δυναδικού αριθμού Γ, να υπολογίσετε στο δυναδικό σύστημα αρίθμησης, το αποτέλεσμα της **αφαίρεσης B-Γ**.

(Μονάδες 2)

ΑΣΚΗΣΗ 3

(α) Δίνεται το πιο κάτω πρόγραμμα στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας οποιονδήποτε τρόπο (π.χ. μέθοδο της προκαταρκτικής εκτέλεσης), να γράψετε τα αποτελέσματα του προγράμματος, όπως θα τυπωθούν στην οθόνη.

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

int syn3a(int a, int &b){
    int c = 0;
    for(int i=a; i<=b; i+=2)
        c += i;
    a++;
    b++;
    return c;
}

int main(){
    int x = 2, y = 7, z;
    z = syn3a(x,y);
    cout << x << endl;
    cout << y << endl;
    cout << z;
    return 0;
}
```

(Μονάδες 3)

(β) Το πιο κάτω πρόγραμμα, στη γλώσσα προγραμματισμού C++, διαβάζει από το πληκτρολόγιο τη συμβολοσειρά **stA**. Στη συνέχεια καλεί τη συνάρτηση **syn3b** η οποία δέχεται ως παράμετρο τη συμβολοσειρά **stA**. Η συνάρτηση **syn3b** υπολογίζει και επιστρέφει στην κύρια συνάρτηση (**main**) το πλήθος των κεφαλαίων χαρακτήρων του λατινικού αλφαβήτου που υπάρχουν μέσα στη συμβολοσειρά **stA**. Η επιστρεφόμενη τιμή τυπώνεται στην κύρια συνάρτηση.

Στο πρόγραμμα υπάρχουν λογικά ή/και συντακτικά λάθη. Να γράψετε στο τετράδιο απαντήσεών σας **τέσσερα (4)** από αυτά, αναφέροντας τον αριθμό της γραμμής στην οποία εμφανίζεται το κάθε λάθος μαζί με τη διορθωμένη εντολή. Στο πρόγραμμα **να μην γίνει κάποια προσθήκη ή αφαίρεση εντολής**.

```
/*1*/ #include <iostream>
/*2*/ #include <string>
/*3*/ using namespace std;
/*4*/ int syn3b(string a){
/*5*/     int p;
/*6*/     for(int i=0; i<=a.size(); i++)
/*7*/         if(a[i]>='A' || a[i]<='Z')
/*8*/             p++;
/*9*/     return p;
/*10*/ }
/*11*/ int main(){
/*12*/     string stA;
/*13*/     cin >> stA;
/*14*/     cout << syn3b(string stA);
/*15*/     return 0;
/*16*/ }
```

(Μονάδες 2)

ΑΣΚΗΣΗ 4:

Μία έγκυρη διεύθυνση ηλεκτρονικού ταχυδρομείου (email) έχει τη μορφή όνομα_χρήστη@τομέας. Για παράδειγμα, το όνομα χρήστη στο email student@schools.ac.cy είναι η συμβολοσειρά student.

Να γράψετε πρόγραμμα, στη γλώσσα προγραμματισμού C++, το οποίο να διαβάζει από το αρχείο **accounts.txt** τα emails άγνωστου πλήθους φοιτητών και στη συνέχεια να τυπώνει, σε διαφορετικές γραμμές, στο αρχείο **userscodes.txt** τα ονόματα των χρηστών (usernames).

(Μονάδες 5)

Παράδειγμα Εισόδου (accounts.txt)	Παράδειγμα Εξόδου (userscodes.txt)
john@st.schools.ac.cy mary@cs.ouc.cy nick@cs.ucy.cy	john mary nick

ΑΣΚΗΣΗ 5:

Το παραγοντικό ενός θετικού ακέραιου αριθμού N , συμβολίζεται με $N!$ και είναι το γινόμενο όλων των ακεραίων από το 1 μέχρι και το N . Για παράδειγμα: $4! = 1 * 2 * 3 * 4 = 24$.

Να γράψετε πρόγραμμα, στη γλώσσα προγραμματισμού C++, το οποίο να:

(α) **υπολογίζει** το παραγοντικό του κάθε αριθμού από το 1 μέχρι και το 6 και να το καταχωρίζει σε έναν μονοδιάστατο πίνακα με όνομα **F**.

(Μονάδες 2)

(β) διαβάζει έναν θετικό ακέραιο αριθμό **N** ($1 \leq N < 1000$) και χρησιμοποιώντας τις τιμές του πίνακα **F** να εκφράζει τον αριθμό N ως άθροισμα παραγοντικών, επιλέγοντας σε κάθε βήμα το **μεγαλύτερο δυνατό** παραγοντικό από τον πίνακα. Για κάθε παραγοντικό που συμμετέχει στο άθροισμα, να εμφανίζει τον συντελεστή (δηλαδή πόσες φορές χρησιμοποιείται το κάθε παραγοντικό) και το αντίστοιχο παραγοντικό του.

Σημείωση: Να τυπώνονται μόνο τα παραγοντικά που έχουν μη μηδενικούς συντελεστές.

(Μονάδες 3)

Παράδειγμα Εισόδου	Παράδειγμα Εξόδου	Επεξήγηση
13	2*3! 1*1!	$2*3! + 1*1! = 12 + 1 = 13$
181	1*5! 2*4! 2*3! 1*1!	$1*5! + 2*4! + 2*3! + 1*1! = 120 + 48 + 12 + 1 = 181$

ΑΣΚΗΣΗ 6:

Να γράψετε συνάρτηση, στη γλώσσα προγραμματισμού C++, με πρότυπο:

```
int bsearch(int N, string W[], string target);
```

Η συνάρτηση δέχεται ως παραμέτρους:

- έναν ακέραιο αριθμό **N**, ο οποίος αντιστοιχεί στο πλήθος των συμβολοσειρών του πίνακα
- έναν πίνακα **W**, ο οποίος περιέχει **N** διαφορετικές συμβολοσειρές, σε αλφαβητική σειρά
- μια συμβολοσειρά με όνομα **target** η οποία δεν υπάρχει στον πίνακα **W**

Η συνάρτηση να χρησιμοποιεί τον αλγόριθμο **Διαδικής Αναζήτησης (Binary Search)** και να επιστρέφει τη θέση του πίνακα **μετά την οποία** πρέπει να εισαχθεί η συμβολοσειρά **target**, ώστε ο πίνακας να παραμείνει ταξινομημένος.

Αν η συμβολοσειρά **target** πρέπει να τοποθετηθεί πριν από το πρώτο στοιχείο του πίνακα, η συνάρτηση πρέπει να επιστρέφει την τιμή -1.

Παράδειγμα:

Παράμετροι Συνάρτησης	Επιστρέφει	Επεξήγηση
N = 3 Πίνακας W: "aa", "abc", "dog" target: "add"	1	Η συμβολοσειρά "add" πρέπει να τοποθετηθεί μετά τη λέξη "abc" που βρίσκεται στη θέση 1.
N = 3 Πίνακας W: "cat", "dog", "fox" target: "bat"	-1	Η συμβολοσειρά "bat" πρέπει να τοποθετηθεί πριν από το πρώτο στοιχείο του πίνακα.

(Μονάδες 5)

- ΤΕΛΟΣ Α΄ ΜΕΡΟΥΣ -
- ΑΚΟΛΟΥΘΕΙ ΤΟ ΜΕΡΟΣ Β΄ -

ΜΕΡΟΣ Β΄

ΑΣΚΗΣΗ 7:

(α) Το Σύστημα Ανίχνευσης Κόπωσης Οδηγού ενεργοποιεί σταδιακή επιβράδυνση του αυτοκινήτου μέχρι την πλήρη ακινητοποίησή του. Το σύστημα χρησιμοποιεί τους πιο κάτω αισθητήρες:

- **A - Χρόνος Οδήγησης:** βρίσκεται στην κατάσταση 1 όταν ο οδηγός οδηγεί πάνω από τέσσερις συνεχόμενες ώρες, διαφορετικά βρίσκεται στην κατάσταση 0.
- **B - Κάμερα βλεφάρων:** βρίσκεται στην κατάσταση 1 όταν τα βλέφαρα του οδηγού είναι κλειστά για πάνω από τρία δευτερόλεπτα, διαφορετικά βρίσκεται στην κατάσταση 0.
- **C - Αισθητήρας τιμονιού:** βρίσκεται στην κατάσταση 1 όταν ο οδηγός δεν κρατάει το τιμόνι με τα δύο χέρια, διαφορετικά βρίσκεται στην κατάσταση 0.

Το σύστημα ενεργοποιείται ($F=1$) όταν ισχύει **τουλάχιστον μια** από τις ακόλουθες συνθήκες:

- Ο οδηγός οδηγεί πάνω από τέσσερις συνεχόμενες ώρες και τα βλέφαρα του είναι κλειστά για πάνω από τρία δευτερόλεπτα.
- Ο οδηγός **ΔΕΝ** κρατάει το τιμόνι με τα δύο χέρια.

Να δημιουργήσετε τον πίνακα αληθείας για το πιο πάνω αυτοματοποιημένο σύστημα και να γράψετε την αντίστοιχη λογική συνάρτηση F του συστήματος σε μορφή αθροίσματος ελαχιστόρων.

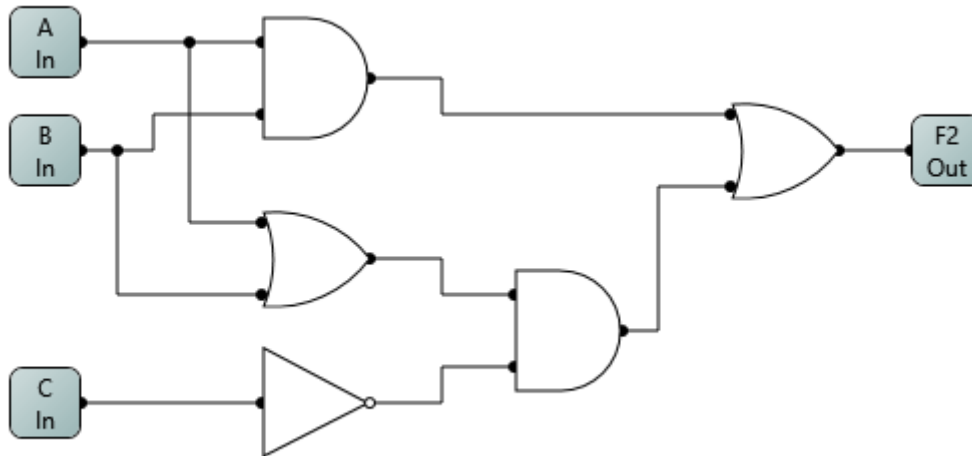
(Μονάδες 5)

(β) Δίνεται ο πιο κάτω χάρτης **Karnaugh** τεσσάρων μεταβλητών. Αφού τον αντιγράψετε στο τετράδιο απαντήσεών σας, να ομαδοποιήσετε τους γειτονικούς του όρους και να γράψετε τη λογική συνάρτηση $F1(A, B, C, D)$ που προκύπτει στην πιο απλή της μορφή.

		CD			
		00	01	11	10
AB	00	0	1	1	0
	01	0	1	1	0
	11	1	1	0	0
	10	1	1	0	0

(Μονάδες 3)

(γ) Να γράψετε τη λογική συνάρτηση **F2** που αντιστοιχεί στο πιο κάτω λογικό κύκλωμα και να υπολογίσετε την τιμή της **F2** αν **A=1**, **B=0** και **C=0**.



(Μονάδες 2)

ΑΣΚΗΣΗ 8:

Δίνεται μια συμβολοσειρά **st** που αποτελείται το πολύ από **1000** λέξεις. Οι λέξεις χωρίζονται με ένα κενό διάστημα. Μετά την τελευταία λέξη δεν υπάρχει κενό διάστημα.

Να γράψετε στη γλώσσα προγραμματισμού C++, τρεις (3) συναρτήσεις όπως αυτές περιγράφονται πιο κάτω, θεωρώντας ότι υπάρχουν δηλωμένες οι απαραίτητες βιβλιοθήκες:

(α) Τη συνάρτηση **count**, η οποία δέχεται ως παράμετρο τη συμβολοσειρά **st** και επιστρέφει το πλήθος των λέξεών της.

Παράδειγμα (συμβολοσειρά st) modify this text for the students	Επιστρέφει 6
--	------------------------

(Μονάδες 2)

(β) Τη συνάρτηση **lexeis**, η οποία δέχεται ως παράμετρο τη συμβολοσειρά **st** και έναν πίνακα συμβολοσειρών με όνομα **words**. Η συνάρτηση χωρίζει τη συμβολοσειρά σε λέξεις και τις αποθηκεύει στον πίνακα **words**.

Παράδειγμα (συμβολοσειρά st) modify this text for the students	Περιεχόμενα πίνακα words modify this text for the students
--	---

(Μονάδες 4)

(γ) Τη συνάρτηση **align**, η οποία δέχεται τις ακόλουθες παραμέτρους:

- τον πίνακα συμβολοσειρών **words** που περιέχει τις λέξεις της συμβολοσειράς
- έναν ακέραιο αριθμό **N** που αντιστοιχεί στο πλήθος των λέξεων που υπάρχουν μέσα στον πίνακα words
- έναν ακέραιο αριθμό **X** που αντιστοιχεί στο μέγιστο πλήθος χαρακτήρων που μπορούν να τυπωθούν σε μια γραμμή

Η συνάρτηση **τυπώνει στην οθόνη** σε γραμμές μεγέθους μέχρι X χαρακτήρες τις λέξεις που υπάρχουν μέσα στον πίνακα. Να θεωρήσετε ότι η κάθε λέξη αποτελείται το πολύ από X χαρακτήρες. Οι λέξεις τυπώνονται με αριστερή στοίχιση και χωρίζονται μεταξύ τους από έναν κενό χαρακτήρα.

Παράδειγμα			
Πίνακας words	Ακέραιος N	Ακέραιος X	Οθόνη
modify this text for the students	6	11	modify this text for the students

(Μονάδες 4)

ΑΣΚΗΣΗ 9:

Η διεύθυνση ενός ταξιδιωτικού γραφείου, επιθυμεί τη δημιουργία προγράμματος, στη γλώσσα προγραμματισμού C++, το οποίο να:

(α) ορίζει μια **εγγραφή** με το όνομα **sights**, η οποία να έχει τα παρακάτω μέλη:

- **name**: το όνομα του αξιοθέατου (συμβολοσειρά χωρίς κενά)
- **town**: την πόλη που βρίσκεται το αξιοθέατο (συμβολοσειρά χωρίς κενά)
- **V**: αριθμός επισκεπτών τα τελευταία επτά (7) χρόνια (πίνακας ακεραίων)

Στη συνέχεια να **ορίζει** έναν **πίνακα εγγραφών** τύπου **sights**, με όνομα **S**, πενήντα (50) θέσεων.

(Μονάδες 2)

(β) διαβάζει τα στοιχεία: όνομα αξιοθέατου, πόλη και αριθμός επισκεπτών τα τελευταία 7 χρόνια για κάθε αξιοθέατο και καταχωρίζει τα στοιχεία αυτά στον πίνακα εγγραφών **S**. Η είσοδος δεδομένων τερματίζεται όταν ο χρήστης καταχωρίσει στοιχεία για 50 αξιοθέατα ή όταν δώσει ως όνομα αξιοθέατου τη λέξη «EXIT».

(Μονάδες 4)

(γ) εμφανίζει στην οθόνη το **όνομα** και την **πόλη** των αξιοθέατων για τα οποία ο αριθμός των επισκεπτών παρουσιάζει συνεχή ανοδική πορεία τα τελευταία 7 χρόνια. Δηλαδή, ο αριθμός των επισκεπτών κάθε χρονιάς είναι μεγαλύτερος από τον αντίστοιχο της αμέσως προηγούμενης χρονιάς.

(Μονάδες 4)

(Ακολουθεί παράδειγμα στην επόμενη σελίδα)

Παράδειγμα Εισόδου	Παράδειγμα Εξόδου
Κάστρο_Λεμεσού Λεμεσός 1200 1350 1410 1500 1620 1750 1900 Τάφοι_των_Βασιλέων Πάφος 2500 2600 2550 2700 2850 3000 3150 Μουσείο_Κύπρου Λευκωσία 1800 1950 2100 2250 2400 2600 2800 EXIT	Κάστρο_Λεμεσού - Λεμεσός Μουσείο_Κύπρου - Λευκωσία

ΑΣΚΗΣΗ 10:

Οι γονείς μιας μαθήτριας της χάρισαν **K** κουτιά με σοκολάτες. Κάθε κουτί περιέχει διαφορετικό είδος σοκολατών. Δύο ή περισσότερα κουτιά μπορεί να περιέχουν την ίδια ποσότητα. Κάθε κουτί περιέχει από 1 μέχρι και 50 σοκολάτες. Η μαθήτρια τρώει μία σοκολάτα την ημέρα για τις πέντε (5) μέρες που πηγαίνει στο σχολείο (Δευτέρα έως Παρασκευή).

Να γράψετε πρόγραμμα στη γλώσσα προγραμματισμού C++, το οποίο να:

- (α) διαβάζει έναν ακέραιο αριθμό **K** ($1 \leq K \leq 500$) ο οποίος αντιπροσωπεύει το πλήθος των κουτιών σοκολάτας. Ακολουθως, να διαβάζει το πλήθος σοκολατών για κάθε ένα από τα **K** κουτιά και να τα καταχωρίζει στον πίνακα **chocos**.

(Μονάδες 2)

- (β) εντοπίζει και τυπώνει την ποσότητα σοκολατών (**F**) που εμφανίζεται τις περισσότερες φορές αρχικά στα κουτιά. Να θεωρήσετε ότι υπάρχει μόνο μία τέτοια ποσότητα.

(Μονάδες 3)

- (γ) υπολογίζει και τυπώνει τον συνολικό αριθμό ημερών (**D**) κατά τις οποίες η μαθήτρια τρώει σοκολάτες, δεδομένου ότι ακολουθεί τους εξής κανόνες:

- Κάθε μέρα τρώει ακριβώς μία σοκολάτα.
- Κατά τη διάρκεια μιας εβδομάδας (**πέντε μέρες**) δεν μπορεί να φάει σοκολάτα από το ίδιο κουτί. Σε διαφορετικές εβδομάδες μπορεί να φάει ξανά σοκολάτα από το ίδιο κουτί.
- Η μαθήτρια επιλέγει τις σοκολάτες που τρώει με τον καλύτερο δυνατό τρόπο, ώστε το πλήθος των ημερών να είναι το μέγιστο δυνατό, δηλαδή να φάει όσον το δυνατό περισσότερες σοκολάτες.
- Σταματά να τρώει σοκολάτες την ημέρα κατά την οποία δεν υπάρχει διαθέσιμο διαφορετικό είδος σοκολάτας για να φάει, εντός της ίδιας εβδομάδας.

(Μονάδες 5)

(Ακολουθεί παράδειγμα στην επόμενη σελίδα)

Παράδειγμα Εισόδου	Παράδειγμα Εξόδου	Επεξήγηση
5 1 1 1 1 2	F: 1 D: 6	<p>Η ποσότητα που εμφανίζεται τις περισσότερες φορές είναι η 1 (4 φορές).</p> <p>Την 1^η εβδομάδα τρώει από τα 5 διαφορετικά κουτιά σοκολατών. Τη 2^η εβδομάδα μόνο το 5^ο κουτί έχει ακόμη διαθέσιμη ποσότητα, άρα τρώει ακόμη μια σοκολάτα.</p>
6 3 5 3 2 4 1	F: 3 D: 17	<p>Η ποσότητα που εμφανίζεται τις περισσότερες φορές είναι η 3 (2 φορές).</p> <p>Με βέλτιστη επιλογή σοκολατών: Τρεις εβδομάδες από 5 διαφορετικά κουτιά σοκολατών και την 4^η εβδομάδα άλλες 2 σοκολάτες.</p>
4 30 30 30 50	F: 30 D: 4	<p>Η ποσότητα που εμφανίζεται τις περισσότερες φορές είναι η 30 (3 φορές).</p> <p>Υπάρχουν μόνο 4 διαφορετικά κουτιά σοκολατών, οπότε την 5^η ημέρα της 1^{ης} εβδομάδας σταματά γιατί δεν υπάρχει διαθέσιμο άλλο είδος σοκολάτας.</p>

- ΤΕΛΟΣ Β΄ ΜΕΡΟΥΣ -
- ΑΚΟΛΟΥΘΕΙ ΤΟ ΜΕΡΟΣ Γ΄ -

ΜΕΡΟΣ Γ΄

Άσκηση 11:

Μια ομάδα χωραφιών είναι διατεταγμένη σε ένα δισδιάστατο πλέγμα με **N** γραμμές και **M** στήλες ($5 \leq N < 1000$, $5 \leq M < 1000$). Μετά από έντονη βροχόπτωση, ορισμένα χωράφια έχουν πλημμυρίσει. Σε κάθε γραμμή αρχικά μπορεί να υπάρχει το πολύ ένα πλημμυρισμένο χωράφι.

Δίνεται το πιο κάτω τμήμα προγράμματος στη γλώσσα προγραμματισμού C++:

```
int main() {
    char A[1000][1000];
    int B[1000], F[1000], N, M;
    cin >> N >> M;
    fill(A, N, M);
    calc(A, B, F, N, M);
    cout << "D: " << dry(B, F, N, M) << endl;
    cout << "T: " << printT(B, F, N, M);
    return 0;
}
```

(α) Να γράψετε τη συνάρτηση **fill**, η οποία διαβάζει τα στοιχεία του πλέγματος και τα καταχωρίζει στον δισδιάστατο πίνακα **A**.

Κάθε θέση του πίνακα περιέχει έναν από τους ακόλουθους χαρακτήρες:

- **O**: στεγνό χωράφι
- **X**: πλημμυρισμένο χωράφι (το πολύ ένα σε κάθε γραμμή)
- **#**: ανάχωμα (το πολύ ένα σε κάθε γραμμή)

(Μονάδες 2)

(β) Να γράψετε τη συνάρτηση **calc** η οποία υπολογίζει και καταχωρίζει στον παράλληλο μονοδιάστατο πίνακα **B** τη θέση του αναχώματος κάθε γραμμής (δηλαδή τον δείκτη της στήλης μέσα στην οποία βρίσκεται). Επίσης, υπολογίζει και καταχωρίζει στον παράλληλο μονοδιάστατο πίνακα **F** τη θέση του χωραφιού που έχει πλημμυρίσει. Αν σε μια γραμμή δεν υπάρχει ανάχωμα καταχωρίζεται στον πίνακα B η τιμή -1. Αν σε μια γραμμή δεν υπάρχει πλημμυρισμένο χωράφι να καταχωρίζεται στον πίνακα F η τιμή -1.

(Μονάδες 3)

(γ) Να γράψετε τη συνάρτηση **dry** η οποία υπολογίζει και επιστρέφει το πλήθος των χωραφιών (**D**) που θα παραμείνουν στεγνά μετά την πλήρη εξάπλωση της πλημμύρας. Το νερό εξαπλώνεται μόνο οριζόντια μέσα στην ίδια γραμμή. Αν ένα χωράφι είναι πλημμυρισμένο, τότε όλα τα χωράφια στο ίδιο συνεχόμενο οριζόντιο τμήμα της γραμμής πλημμυρίζουν, εκτός αν παρεμβάλλεται ανάχωμα. Δηλαδή, η ροή του νερού σταματά όταν συναντήσει ανάχωμα. Τα αναχώματα δεν υπολογίζονται ως χωράφια.

(Μονάδες 5)

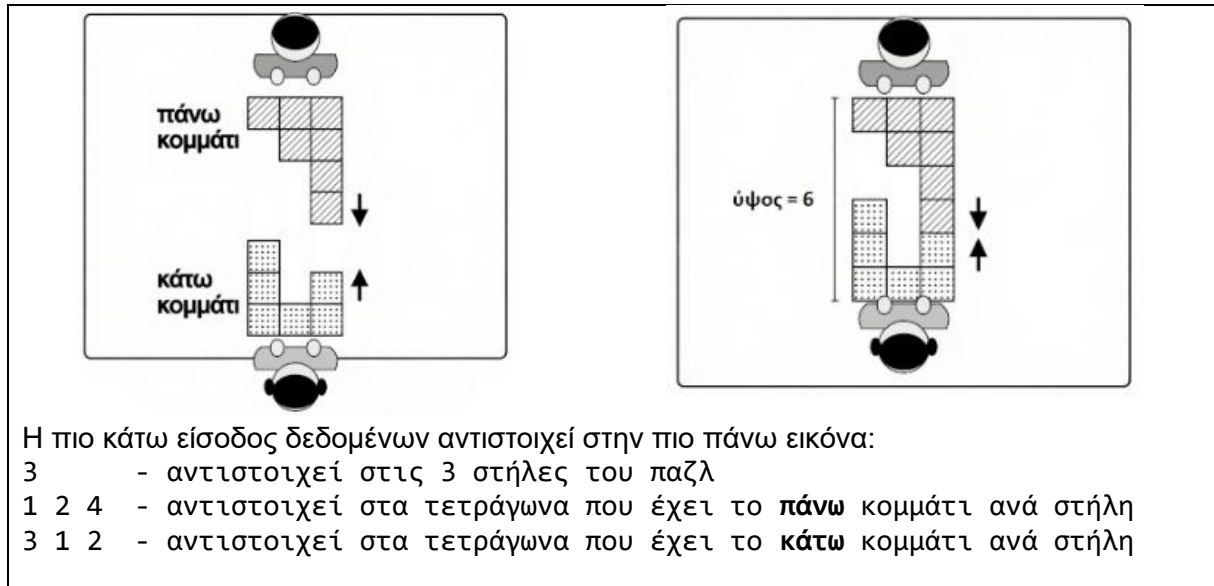
(δ) Να γράψετε τη συνάρτηση **printT** που επιστρέφει τον μέγιστο αριθμό χωραφιών (**T**) που μπορούν να παραμείνουν στεγνά, **ενώ προηγουμένως θα πλημμύριζαν**, αν τοποθετηθούν στις κατάλληλες θέσεις επιπρόσθετα αναχώματα, στο **αρχικό πλέγμα**. Τα αναχώματα δεν τοποθετούνται στη θέση των πλημμυρισμένων χωραφιών. Σε κάθε γραμμή μπορεί να υπάρχει μόνο ένα ανάχωμα. Τα αναχώματα δεν υπολογίζονται ως χωράφια.

(Μονάδες 5)

Παράδειγμα Εισόδου	Παράδειγμα Εξόδου	Επεξήγηση
<pre> 8 7 ○ X ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ X ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ X ○ ○ ○ ○ X ○ # ○ ○ X ○ ○ ○ ○ ○ ○ ○ # ○ ○ ○ X ○ # ○ ○ ○ ○ </pre>	<pre> D: 17 T: 16 </pre>	<p>Μετά την εξάπλωση του νερού (ερώτημα γ) τα χωράφια έχουν την εξής μορφή:</p> <pre> X X X X X X X X X X X X X X ○ ○ ○ ○ ○ ○ ○ X X X X X X X X X X X X X # X X X X X X X ○ ○ ○ # ○ ○ ○ X X # ○ ○ ○ ○ </pre> <p>Μετά την προσθήκη των επιπρόσθετων αναχωμάτων στο αρχικό πλέγμα και την εξάπλωση του νερού (ερώτημα δ) τα χωράφια έχουν την εξής μορφή:</p> <pre> X X # ○ ○ ○ ○ ○ ○ ○ ○ # X X ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ # X X X X X X X # X X X # ○ ○ ○ ○ ○ ○ # ○ ○ ○ X X # ○ ○ ○ ○ </pre>

ΑΣΚΗΣΗ 12:

Δύο παιδιά συνεργάζονται για να φτιάξουν ένα παζλ N στηλών αποτελούμενο από δύο (2) κομμάτια. Κάθονται απέναντι σε ένα τραπέζι και το κάθε κομμάτι ακουμπάει την άκρη του τραπεζιού (βλέπε εικόνες). Μόλις τελειώσει κάθε παιδί το δικό του κομμάτι, σπρώχνουν και τα δύο κομμάτια μέχρι αυτά να **ακουμπήσουν μεταξύ τους τουλάχιστον σε μια στήλη**.



Να γράψετε πρόγραμμα, στη γλώσσα προγραμματισμού C++, το οποίο να:

(α) διαβάζει έναν ακέραιο αριθμό N ($2 \leq N \leq 16$), ο οποίος δηλώνει το πλήθος των στηλών του παζλ. Στη συνέχεια, να καταχωρίζει στον μονοδιάστατο πίνακα **top** το πλήθος των τετραγώνων για κάθε μια από τις N στήλες του πάνω κομματιού και ακολούθως, στον μονοδιάστατο πίνακα **bot** το πλήθος των τετραγώνων για κάθε μια από τις N στήλες του κάτω κομματιού.

(Μονάδες 2)

(β) υπολογίζει και εμφανίζει το μέγιστο ύψος (H) του παζλ.

(Μονάδες 4)

(γ) υπολογίζει και εμφανίζει το συνολικό πλήθος (E) των άδειων σημείων ενδιάμεσα των δύο κομματιών.

(Μονάδες 4)

(δ) υπολογίζει και εμφανίζει το πλήθος των κενών σημείων (F) από τα οποία αποτελείται η αριστερότερη κενή περιοχή (δύο κενά σημεία ανήκουν στην ίδια περιοχή αν έχουν κοινή πλευρά).

(Μονάδες 5)

(Ακολουθεί παράδειγμα στην επόμενη σελίδα)

Παράδειγμα Εισόδου	Παράδειγμα Εξόδου	Επεξήγηση
3 1 2 4 3 1 2	H:6 E:5 F:5	<pre> T T T . T T . . T B . T B . B B B B </pre>
2 1 1 1 1	H:2 E:0 F:0	<pre> T T B B </pre>
6 1 2 1 1 5 1 1 3 1 2 1 3	H:6 E:14 F:12	<pre> T T T T T T . T . . T T . . B . . T B . B . B T B B B B B B B </pre>
4 5 1 2 3 2 5 2 1	H:7 E:7 F:1	<pre> T T T T T . T T T B . T T B . . T B . . B B B . B B B B </pre>

Σημείωση: Με τον χαρακτήρα 'T' εμφανίζονται τα τετράγωνα του πάνω κομματιού, με τον χαρακτήρα 'B' εμφανίζονται τα τετράγωνα του κάτω κομματιού και με τον χαρακτήρα '.' τα κενά σημεία. Η αριστερότερη κενή περιοχή εμφανίζεται με περίγραμμα.

- ΤΕΛΟΣ Γ' ΜΕΡΟΥΣ -

- ΤΕΛΟΣ ΕΞΕΤΑΣΗΣ -

**ΤΥΠΟΛΟΓΙΟ ΣΥΝΑΡΤΗΣΕΩΝ ΚΑΙ ΣΤΑΘΕΡΩΝ ΣΤΗ ΓΛΩΣΣΑ
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C++**

ΣΥΝΑΡΤΗΣΕΙΣ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ <cmath>		
Συνάρτηση	Χρήση	Παράμετροι
sqrt(x)	Επιστρέφει την τετραγωνική ρίζα του αριθμού x. Η επιστρεφόμενη τιμή είναι πραγματικός αριθμός.	Ένας θετικός αριθμός (ακέραιος ή πραγματικός)
abs(x)	Επιστρέφει την απόλυτη τιμή του αριθμού x. Η επιστρεφόμενη τιμή εξαρτάται από τον τύπο του αριθμού x.	Ένας αριθμός (ακέραιος ή πραγματικός)
pow(x,y)	Επιστρέφει το αποτέλεσμα της δύναμης x^y . Η επιστρεφόμενη τιμή είναι πραγματικός αριθμός.	Δύο πραγματικοί αριθμοί
trunc(x)	Επιστρέφει το ακέραιο μέρος του αριθμού x σε πραγματική μορφή, αγνοώντας το δεκαδικό μέρος του .	Ένας πραγματικός αριθμός
round(x)	Επιστρέφει το ακέραιο μέρος του αριθμού x σε πραγματική μορφή, στρογγυλοποιημένο στην πλησιέστερη τιμή .	Ένας πραγματικός αριθμός
ΣΥΝΑΡΤΗΣΕΙΣ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ <string>		
size()	Επιστρέφει το μέγεθος μιας συμβολοσειράς. Η επιστρεφόμενη τιμή είναι ακέραιος αριθμός που συμβολίζει από πόσα bytes αποτελείται μια συμβολοσειρά.	Καμία παράμετρος
clear()	Διαγράφει το περιεχόμενο μιας συμβολοσειράς. Δεν επιστρέφει τίποτα.	Καμία παράμετρος
empty()	Ελέγχει αν μια συμβολοσειρά είναι άδεια . Η επιστρεφόμενη τιμή είναι τύπου Boolean .	Καμία παράμετρος
getline(x,y)	Αποθηκεύει ολόκληρη συμβολοσειρά που μπορεί να εισαχθεί από το πληκτρολόγιο ή από αρχείο (x) στο αντικείμενο y.	Η 1 ^η παράμετρος (x) αφορά στη μέθοδο εισαγωγής της συμβολοσειράς (από το πληκτρολόγιο ή από αρχείο) και η 2 ^η παράμετρος (y) αφορά στο αντικείμενο στο οποίο θα αποθηκευτεί η συμβολοσειρά η οποία εισάγεται.
ΣΤΑΘΕΡΕΣ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ <climits>		
INT_MAX	Μέγιστο αριθμητικό όριο μεταβλητής ή σταθεράς τύπου integer . Η ακριβής αριθμητική τιμή της είναι 32767 (στα 2 bytes) ή 2147483647 (στα 4 bytes)	
INT_MIN	Ελάχιστο αριθμητικό όριο μεταβλητής ή σταθεράς τύπου integer . Η ακριβής αριθμητική τιμή της είναι -32768 (στα 2 bytes) ή -2147483648 (στα 4 bytes)	