

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΑΘΛΗΤΙΣΜΟΥ
ΚΑΙ ΝΕΟΛΑΙΑΣ
ΔΙΕΥΘΥΝΣΗ ΑΝΩΤΕΡΗΣ ΕΚΠΑΙΔΕΥΣΗΣ
ΥΠΗΡΕΣΙΑ ΕΞΕΤΑΣΕΩΝ

ΠΑΓΚΥΠΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΠΡΟΣΒΑΣΗΣ 2024

Μάθημα: ΠΛΗΡΟΦΟΡΙΚΗ (15)

Ημερομηνία και ώρα εξέτασης: Τετάρτη, 19 Ιουνίου 2024

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.
- 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 Διαγραμμάτων και των Λογικών Κυκλωμάτων, καθώς και το λεκτικό περιεχόμενό τους μπορούν να γίνουν με μολύβι.

ΜΕΡΟΣ Α'

ΑΣΚΗΣΗ 1:

Το Υπουργείο Ενέργειας, Εμπορίου και Βιομηχανίας επιχορηγεί την εγκατάσταση φωτοβολταϊκού συστήματος σε οικιακούς καταναλωτές βάσει εισοδηματικών κριτηρίων ως ακολούθως:

Αν το **μηνιαίο εισόδημα** του καταναλωτή είναι **μικρότερο** από **€1200** τότε δίνεται επιχορήγηση **60%** πάνω στο **αρχικό κόστος**, διαφορετικά δίνεται επιχορήγηση **20%**.

Να σχεδιάσετε **λογικό διάγραμμα**, το οποίο:

(α) να **δέχεται** το **μηνιαίο εισόδημα** του οικιακού καταναλωτή και το **αρχικό κόστος** του φωτοβολταϊκού συστήματος.

(Μονάδες 1)

(β) να **υπολογίζει** το **ποσό επιχορήγησης** και το **τελικό κόστος** του φωτοβολταϊκού συστήματος.

(Μονάδες 3)

(γ) να **τυπώνει** το **τελικό κόστος** και το **ποσό επιχορήγησης** του φωτοβολταϊκού συστήματος, όπως αυτό έχει υπολογιστεί στο ερώτημα (β).

(Μονάδες 1)

ΑΣΚΗΣΗ 2:

Δίνονται στο **δεκαδικό** σύστημα αρίθμησης ο πραγματικός αριθμός **A=8.375** και στο **δυναδικό** σύστημα αρίθμησης οι αριθμοί **B=01110001** και **Γ=00101010**.

(α) Να δείξετε ότι η αντίστοιχη τιμή του **πραγματικού αριθμού A** στο **δυναδικό** σύστημα αρίθμησης είναι **(1000.011)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 **(113)₁₀**, **σημειώνοντας τα βήματα** που ακολουθήσατε για να φτάσετε στο συγκεκριμένο αποτέλεσμα.

(Μονάδες 1)

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

(Μονάδες 2)

ΑΣΚΗΣΗ 3:

Δίνεται το πιο κάτω πρόγραμμα στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης, να παρουσιάσετε τα αποτελέσματα του προγράμματος. Στη θέση του κενού διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```

#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;
int test(int a,int &b){
    int c;
    if(a>b)
        a=b*2;
    else
        a=b*3;
    b=a-b;
    cout<<setw(3)<<a<<setw(3)<<b<<endl;
    c=trunc((a+b)*0.5);
    return c;
}
int main(){
    int x=1,y=2,z;
    z=test(x,y);
    cout<<x<<' '<<y<<endl;
    y=test(z,x);
    cout<<x<<' '<<y<<endl;
return 0;
}

```

(Μονάδες 5)

ΑΣΚΗΣΗ 4:

(α) Να μετατρέψετε τις πιο κάτω **λεκτικές προτάσεις** στις αντίστοιχες **λογικές εκφράσεις** στη γλώσσα προγραμματισμού C++:

- i) το **ψηφίο** των **μονάδων** του αριθμού **num** να είναι **άνισο** του **5**.
- ii) η συμβολοσειρά **str1** να είναι **άδεια** και ο **τελευταίος** χαρακτήρας της συμβολοσειράς **str2** να είναι ο χαρακτήρας **'S'**.

(Μονάδες 1)

(β) Οι μεταβλητές **a**, **b** και **c** είναι τύπου **integer** και έχουν τις ακόλουθες τιμές:
a=2, **b=5** και **c=-2**.

Να γράψετε τις τιμές που θα έχουν οι λογικές μεταβλητές **x** και **y** (boolean), όταν εκτελεστούν οι πιο κάτω εντολές στη γλώσσα προγραμματισμού C++:

- i) `x = b-c<4 && round((b*a-1)/4.0)==5%6;`
- ii) `y = pow(b,a)-20>b || b%c==1 && b/c<2;`

(Μονάδες 2)

(γ) Το πιο κάτω πρόγραμμα στη γλώσσα προγραμματισμού C++ **δέχεται** από τον χρήστη στοιχεία **50** κρασιών. Τα στοιχεία αυτά συμπεριλαμβάνονται σε μια **εγγραφή** με όνομα **cava** και είναι: το **όνομα** του κρασιού, το **έτος** παραγωγής του και η **τιμή πώλησης** του. Ακολούθως, **υπολογίζει** και **τυπώνει** το **όνομα** και το **έτος** των κρασιών που έχουν παραχθεί **πριν** το **2010** και η **τιμή πώλησής** τους είναι **πάνω** από **100** ευρώ. Στην περίπτωση που **δεν υπάρχει τέτοιο κρασί**, τυπώνεται το μήνυμα **«Δεν υπάρχουν τέτοια κρασιά!»**.

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 μη γίνει οποιαδήποτε προσθήκη ή αφαίρεση εντολής.**

```
/*1*/ #include<iostream>
/*2*/ #include<iomanip>
/*3*/ using namespace std;
/*4*/ #define n 50
/*5*/ struct cava{
/*6*/     string name;
/*7*/     int etos;
/*8*/     float timi;
/*9*/ }
/*10*/ int main(){
/*11*/     int i;
/*12*/     bool found=false;
/*13*/     wine cava[n];
/*14*/     for(i=0;i<n;i++)
/*15*/         cin>>wine[i].name>>wine[i].etos>>wine[i].timi;
/*16*/     for(i=0;i<n;i++)
/*17*/         if(wine[i].etos>2010 && wine[i].timi>100){
/*18*/             cout<<wine[i].name<<" "<<wine[i].etos<<endl;
/*19*/             found=true;
/*20*/         }
/*21*/     if(found==true)
/*22*/         cout<<"Δεν υπάρχουν τέτοια κρασιά!"<<endl;
/*23*/     return 0;
/*24*/ }
```

(Μονάδες 2)

ΑΣΚΗΣΗ 5:

Σας δίνεται το πιο κάτω πρόγραμμα στη γλώσσα προγραμματισμού C++ το οποίο καταχωρίζει **τριάντα (30) ονόματα** εβδομαδιαίων εφημερίδων και την αντίστοιχη **τιμή πώλησης** της κάθε εφημερίδας σε **μονοδιάστατους παράλληλους** πίνακες **30 θέσεων** με ονόματα **name** και **timi**, αντίστοιχα. Ακολούθως, ταξινομεί τους δύο πίνακες σε **αύξουσα** σειρά, με βάση την **τιμή** της κάθε εφημερίδας, χρησιμοποιώντας τον αλγόριθμο **ταξινόμησης της Φυσαλίδας (Bubble sort)**. Τέλος, **υπολογίζει** και **τυπώνει** στο **τέλος** του αρχείου **out5.txt** τη **διαφορά** της **ψηλότερης** και της **χαμηλότερης** τιμής.

```
#include<iostream>
#include <fstream>
#define N 30
using namespace std;
```

A

```
int main(){
    int i;
    float temp1,temp2;
    string name[N],timi[N];
    bool sorted;
    for (i=0;i<N;i++)
        cin>>name[i]>>timi[i];
    do{
        sorted=true;
        for (i=0;i<N-1;i++){
            B
        }
    }while (sorted==false);
    Γ
    return 0;
}
```

Να γράψετε στο τετράδιο απαντήσεών σας:

- (α) την **εντολή** που πρέπει να τοποθετηθεί στη **θέση A** για τη δημιουργία μιας **ροής εγγραφής** με το όνομα **fout** έτσι ώστε η διαφορά των **τιμών** των εφημερίδων να **τυπώνεται** στο **τέλος** του αρχείου **out5.txt**.
(Μονάδες 1)
- (β) τις **εντολές** που πρέπει να τοποθετηθούν στη **θέση B**, έτσι ώστε οι δύο πίνακες **name** και **timi** να **ταξινομούνται** σε **αύξουσα** σειρά, με βάση την **τιμή** της κάθε εφημερίδας, χρησιμοποιώντας τον αλγόριθμο **ταξινόμησης της Φυσαλίδας (Bubble sort)**.
(Μονάδες 2)
- (γ) την/τις **εντολή/εντολές** που πρέπει να τοποθετηθεί/τοποθετηθούν στη **θέση Γ**, έτσι ώστε να **υπολογίζεται** και να **τυπώνεται** στο **τέλος** του αρχείου **out5.txt** η **διαφορά** της **ψηλότερης** και της **χαμηλότερης** τιμής των εφημερίδων.
(Μονάδες 2)

ΑΣΚΗΣΗ 6:

Να γράψετε στη γλώσσα προγραμματισμού C++ μια **συνάρτηση** με όνομα **search**, που να δέχεται παραμετρικά έναν πίνακα **πενήντα (50)** ακέραιων αριθμών με το όνομα **nums** και δύο ακέραιους αριθμούς **A** και **B**. Να **θεωρήσετε** ότι τα στοιχεία του πίνακα **nums** είναι ταξινομημένα σε **αύξουσα σειρά** και οι αριθμοί **A** και **B** **υπάρχουν** στον πίνακα.

Η συνάρτηση να αναζητεί τη **θέση** του **αριθμού A** στον πίνακα **nums** χρησιμοποιώντας τον αλγόριθμο της **Διαδικής αναζήτησης (Binary search)**. Αν κατά τη διαδικασία εντοπισμού του αριθμού **A** εντοπίσει πρώτα τον αριθμό **B**, τότε επιστρέφει τη **θέση** του **αριθμού B**, διαφορετικά επιστρέφει τη **θέση** του **αριθμού A**.

Παράδειγμα: (για πίνακα δέκα (10) αριθμών μόνο)

5	12	53	61	68	90	103	123	231	349
---	----	----	----	----	----	-----	-----	-----	-----

- Αν **A=231** και **B=123**, η συνάρτηση θα εντοπίσει πρώτα τον αριθμό **B** και θα επιστρέψει τη θέση του αριθμού **B**, που είναι η **θέση 7**.
- Αν **A=231** και **B=61**, η συνάρτηση δεν θα εντοπίσει τον αριθμό **B** και θα επιστρέψει τη θέση του αριθμού **A**, που είναι η **θέση 8**.

(Μονάδες 5)

**ΤΕΛΟΣ ΜΕΡΟΥΣ Α'
ΑΚΟΛΟΥΘΕΙ ΤΟ ΜΕΡΟΣ Β'**

ΜΕΡΟΣ Β'

ΑΣΚΗΣΗ 7:

Ένας γεωργός επιθυμεί να αυτοματοποιήσει τη λειτουργία της αντλίας νερού που έχει στο περιβόλι του. Η αντλία έχει τη δυνατότητα να διαρρέεται από ηλεκτρικό ρεύμα απευθείας από την ΑΗΚ ή μέσω συστήματος φωτοβολταϊκών πλαισίων και αντλεί νερό από δύο δεξαμενές νερού. Το σύστημα αυτοματοποίησης αποτελείται από τέσσερις (4) εισόδους ως εξής:

Είσοδος Α: Ελέγχει αν η αντλία διαρρέεται από ηλεκτρικό ρεύμα απευθείας από την ΑΗΚ ή μέσω συστήματος φωτοβολταϊκών πλαισίων (**1 – ηλεκτρικό ρεύμα από ΑΗΚ, 0 – ηλεκτρικό ρεύμα από σύστημα φωτοβολταϊκών πλαισίων**).

Είσοδος Β: Ελέγχει αν είναι πρωί ή βράδυ (**1 – πρωί, 0 – βράδυ**).

Είσοδος C: Ελέγχει αν η υγρασία του εδάφους βρίσκεται **πάνω** ή **κάτω** από το ελάχιστο επιτρεπτό όριο (**1 – πάνω, 0 – κάτω**).

Είσοδος D: Επιλέγει από ποια δεξαμενή να αντλείται το νερό (**1 – Επιλογή δεξαμενής Α, 0 – Επιλογή δεξαμενής Β**). Να θεωρήσετε ότι και οι δύο δεξαμενές (Α και Β) έχουν **πάντοτε** νερό.

Ο γεωργός επιθυμεί να ξεκινά η αντλία νερού για να ποτίζει το περιβόλι του στις πιο κάτω δύο περιπτώσεις:

- όταν διαρρέεται από ρεύμα από **σύστημα φωτοβολταϊκών πλαισίων**, είναι **πρωί**, η **υγρασία** είναι **κάτω** από το ελάχιστο επιτρεπτό όριο και το νερό να αντλείται από **οποιαδήποτε δεξαμενή**.
- όταν διαρρέεται από ρεύμα απευθείας από την ΑΗΚ, είναι **βράδυ**, η **υγρασία** είναι **κάτω** από το ελάχιστο επιτρεπτό όριο και το νερό να αντλείται από **οποιαδήποτε δεξαμενή**.

(α) Να δημιουργήσετε τον **πίνακα αληθείας τεσσάρων (4)** μεταβλητών που αντιστοιχούν στις πιο πάνω εισόδους (**A, B, C, D**) και να **γράψετε** την αντίστοιχη **λογική συνάρτηση** για το πιο πάνω σενάριο.

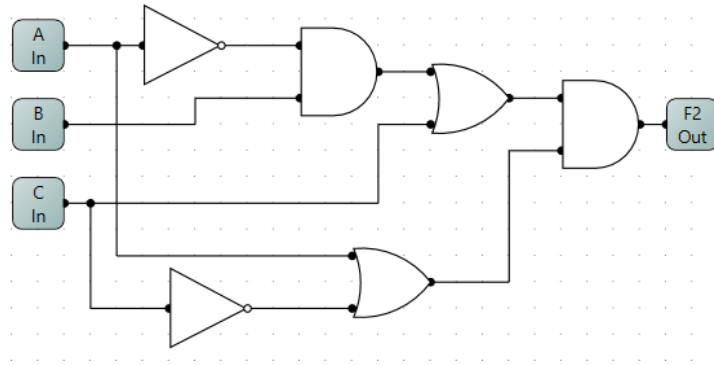
(Μονάδες 5)

(β) Σας δίνεται ο πιο κάτω χάρτης **Karnaugh τεσσάρων (4)** μεταβλητών. Αφού τον **αντιγράψετε** στο τετράδιο απαντήσεών σας, να **ομαδοποιήσετε** τους γειτονικούς του όρους και να **γράψετε** τη **λογική συνάρτηση F1** που προκύπτει στην **πιο απλή της μορφή**.

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

(Μονάδες 3)

(γ) Να **γράψετε** τη λογική **συνάρτηση F2** που αντιστοιχεί στο πιο κάτω λογικό κύκλωμα:



(Μονάδες 2)

ΑΣΚΗΣΗ 8:

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

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

A

```
int main(){
    string str, stright;
    cout<<"Δώσε είκοσι (20)λέξεις:";
```

B

```
return 0;
}
```

Να γράψετε στο τετράδιο απαντήσεών σας:

(α) μια **συνάρτηση** με όνομα **shiftright** που πρέπει να τοποθετηθεί στη **θέση A**, η οποία να **δέχεται** μια **λέξη** και να **μετατοπίζει κυκλικά** τον κάθε χαρακτήρα της κατά **μια** θέση προς τα **δεξιά**, ώστε ο τελευταίος χαρακτήρας της να τοποθετείται στην αρχή. Ακολούθως, να **επιστρέφει** τη νέα λέξη στην κύρια συνάρτηση (main), με τη χρήση **της εντολής return**.

(Μονάδες 6)

(β) τις **εντολές** που πρέπει να τοποθετηθούν στη **θέση B**, έτσι ώστε να **διαβάζει είκοσι (20)** λέξεις, να **καλεί για κάθε μια** από αυτές τη συνάρτηση **shiftright** και να **τυπώνει** την κάθε νέα λέξη που επιστρέφει η συνάρτηση στην οθόνη.

(Μονάδες 4)

Παράδειγμα Εισόδου
(πληκτρολόγιο)

(για 4 λέξεις μόνο)

MOUSE
MONITOR
MOBILE
VARIABLE

Παράδειγμα Εξόδου
(στην οθόνη)

(για 4 λέξεις μόνο)

EMOUS
RMONITO
EMOBIL
EVARIABLE

ΑΣΚΗΣΗ 9:

Ο πατέρας μιας τριμελούς οικογένειας ξέχασε τον **τετραψήφιο νέο** κωδικό της πιστωτικής του κάρτας, ο οποίος έχει επικαιροποιηθεί πρόσφατα. Ο πατέρας θυμάται μόνο τον **παλιό** τετραψήφιο κωδικό που είχε, ενώ η σύζυγός του θυμάται μόνο τα **δύο μεσαία ψηφία** του **νέου** κωδικού του. Επιπλέον, η κόρη τους θυμάται ότι το **άθροισμα** όλων των **ψηφίων** του **παλιού** κωδικού είναι **μεγαλύτερο** κατά **πέντε (5)** μονάδες από το **άθροισμα** όλων των ψηφίων του **νέου** κωδικού.

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

(α) να **διαβάζει** τον **παλιό** κωδικό που θυμάται ο πατέρας και ακολούθως τα **δύο μεσαία ψηφία** του **νέου** κωδικού που θυμάται η σύζυγός του. Να θεωρήσετε ότι τα δεδομένα δίνονται σωστά και δε χρειάζεται οποιοσδήποτε έλεγχος.

(Μονάδες 2)

(β) βάσει των πιο πάνω δεδομένων να **υπολογίζει** και να **τυπώνει όλους** τους **πιθανούς τετραψήφιους κωδικούς** που μπορεί να αντιστοιχούν στον **νέο** κωδικό στο αρχείο **codes.txt**.

(Μονάδες 6)

(γ) να **διαβάζει** όλους τους πιθανούς κωδικούς από το αρχείο **codes.txt**, να **εντοπίζει** και να **τυπώνει** τον **μοναδικό κωδικό** της πιστωτικής κάρτας, αν ο πατέρας ενημερώθηκε από την τράπεζα, ότι το **ψηφίο** των **μονάδων** του **νέου** κωδικού είναι το **ίδιο** με το **ψηφίο** των **χιλιάδων** του **παλιού** κωδικού. Να θεωρήσετε ότι υπάρχει **μόνο ένας** τέτοιος κωδικός στο αρχείο **codes.txt**.

(Μονάδες 2)

Παράδειγμα Εισόδου
(πληκτρολόγιο)

2648
7 4

Παράδειγμα Εξόδου
(στην οθόνη)

Δώσε παλιό κωδικό και τα δύο μεσαία ψηφία του νέου κωδικού:
Νέος κωδικός του πατέρα:2742
(στο αρχείο **codes.txt**)
1743
2742
3741
4740

ΑΣΚΗΣΗ 10:

Η καταγραφή των ελαιόδεντρων για σκοπούς επιχορήγησης και οικονομικής στήριξης των γεωργών από το κράτος γίνεται σε μορφή τετραγωνικού πίνακα **101 γραμμών** και **101 στηλών** με το όνομα **elia**. Κάθε κελί αντιπροσωπεύει ένα διαφορετικό κτήμα με ελαιόδεντρα και περιέχει τον αριθμό των ελαιόδεντρων που υπάρχουν στο κτήμα αυτό. Στο κτήμα που αντιπροσωπεύεται από το κεντρικό κελί του πίνακα δεν υπάρχουν ελαιόδεντρα, αφού εκεί στεγάζεται το ελαιοτριβείο.

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

- (α) να **διαβάζει** και να **καταχωρίζει** στον δισδιάστατο πίνακα με το όνομα **elia** το **πλήθος** των ελαιόδεντρων που υπάρχουν σε κάθε κτήμα. Να θεωρήσετε ότι στο **κεντρικό** κελί καταχωρίζεται η τιμή **-1** και όλα τα υπόλοιπα στοιχεία δίνονται σωστά και δε χρειάζεται οποιοσδήποτε έλεγχος.

(Μονάδες 2)

- (β) λόγω σφάλματος στην αρχική καταχώριση, να **αντιστρέφει** τον πίνακα **elia** και οι τιμές, μετά την αντιστροφή του, να αποθηκεύονται στον πίνακα **eliarev**. Δηλαδή, η 1^η γραμμή να ανταλλάσσεται με την τελευταία γραμμή, με αντεστραμμένες τιμές, η 2^η γραμμή να ανταλλάσσεται με την προτελευταία γραμμή, με αντεστραμμένες τιμές κ.ο.κ. (βλέπε παράδειγμα πιο κάτω). Ακολούθως, να **τυπώνει** τον πίνακα **eliarev** στην **οθόνη**.

Παράδειγμα: (για πίνακα 7 γραμμών και 7 στηλών)

		elia									eliarev						
		0	1	2	3	4	5	6			0	1	2	3	4	5	6
0	77	67	57	47	37	27	16	0	10	21	31	41	51	61	71		
1	76	66	56	46	36	26	15	1 <td>11</td> <td>22</td> <td>32</td> <td>42</td> <td>52</td> <td>62</td> <td>72</td>	11	22	32	42	52	62	72		
2	75	65	55	45	35	25	14	2 <td>12</td> <td>23</td> <td>33</td> <td>43</td> <td>52</td> <td>63</td> <td>73</td>	12	23	33	43	52	63	73		
3	74	64	54	-1	34	24	13	3 <td>13</td> <td>24</td> <td>34</td> <td>-1</td> <td>54</td> <td>64</td> <td>74</td>	13	24	34	-1	54	64	74		
4	73	63	52	43	33	23	12	4 <td>14</td> <td>25</td> <td>35</td> <td>45</td> <td>55</td> <td>65</td> <td>75</td>	14	25	35	45	55	65	75		
5	72	62	52	42	32	22	11	5 <td>15</td> <td>26</td> <td>36</td> <td>46</td> <td>56</td> <td>66</td> <td>76</td>	15	26	36	46	56	66	76		
6	71	61	51	41	31	21	10	6 <td>16</td> <td>27</td> <td>37</td> <td>47</td> <td>57</td> <td>67</td> <td>77</td>	16	27	37	47	57	67	77		

(Μονάδες 4)

- (γ) λόγω επέκτασης του ελαιοτριβείου, να **υπολογίζει** και να **τυπώνει τον αριθμό** των ελαιόδεντρων που πρέπει να αποκοπούν από τα κτήματα που βρίσκονται σε μια τετραγωνική περιοχή, που έχει ως κέντρο το ελαιοτριβείο, στον πίνακα **eliarev**. Η πλευρά (**N**) της τετραγωνικής αυτής περιοχής δίνεται από τον χρήστη. Να θεωρήσετε ότι η πλευρά N είναι **περιττός** αριθμός ($2 < N < 50$) και δεν χρειάζεται οποιοσδήποτε έλεγχος.

Παράδειγμα: (για πίνακα 7 γραμμών και 7 στηλών και N=5)

		eliarev						
		0	1	2	3	4	5	6
0	10	21	31	41	51	61	71	
1	11	22	32	42	52	62	72	
2	12	23	33	43	52	63	73	
3	13	24	34	-1	54	64	74	
4	14	25	35	45	55	65	75	
5	15	26	36	46	56	66	76	
6	16	27	37	47	57	67	77	

Τετραγωνική περιοχή αποκοπής ελαιόδεντρων με πλευρά N=5.
Σύνολο ελαιόδεντρων: 1055

(Μονάδες 4)

**Παράδειγμα Εισόδου
(πληκτρολόγιο)**

(για πίνακα 7 γραμμών και 7 στηλών)

77 67 57 47 37 27 16
76 66 56 46 36 26 15
75 65 55 45 35 25 14
74 64 54 **-1** 34 24 13
73 63 52 43 33 23 12
72 62 52 42 32 22 11
71 61 51 41 31 21 10
5

**Παράδειγμα Εξόδου
(στην οθόνη)**

(για πίνακα 7 γραμμών και 7 στηλών)

Δώσε το πλήθος των ελαιόδεντρων που υπάρχουν σε κάθε κτήμα:

Αντιστροφή του πίνακα elia:

10 21 31 41 51 61 71
11 22 32 42 52 62 72
12 23 33 43 52 63 73
13 24 34 **-1** 54 64 74
14 25 35 45 55 65 75
15 26 36 46 56 66 76
16 27 37 47 57 67 77

Δώσε το μέγεθος της πλευράς N:

Ελαιόδεντρα που πρέπει να αποκοπούν:1055

**ΤΕΛΟΣ ΜΕΡΟΥΣ Β'
ΑΚΟΛΟΥΘΕΙ ΤΟ ΜΕΡΟΣ Γ'**

ΜΕΡΟΣ Γ'

ΑΣΚΗΣΗ 11:

Η εταιρεία ανάπτυξης λογισμικού «SoftDev», που απασχολεί **πεντακόσιους (500)** εργαζόμενους, έχει εξαγοράσει την εταιρεία «DigiTask», που απασχολεί **διακόσιους (200)** εργαζόμενους. Λόγω της συγχώνευσης αποφασίστηκε η ενοποίηση των δύο συστημάτων μισθοδοσίας.

Για την υλοποίηση της πιο πάνω ανάγκης, τα δεδομένα των συστημάτων μισθοδοσίας των δύο εταιρειών έχουν αποθηκευτεί σε δύο ξεχωριστά αρχεία, **softdev.txt** και **digitask.txt** αντίστοιχα. Στα αρχεία αυτά για κάθε υπάλληλο έχουν αποθηκευτεί τα στοιχεία: **όνομα, επίθετο, τμήμα** και **μισθός**.

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

(α) να **διαβάζει** τα δεδομένα των δύο πιο πάνω αρχείων και να τα **καταχωρίζει** σε έναν δισδιάστατο πίνακα **επτακόσιων (700) γραμμών** και **δύο (2) στηλών** με όνομα **data_all** και σε **δύο (2) μονοδιάστατους παράλληλους πίνακες επτακοσίων (700) θέσεων** με ονόματα **tmima_all** και **misthoi_all**.

Στον δισδιάστατο πίνακα **data_all** να καταχωριστούν το **όνομα** και το **επίθετο** του/της κάθε υπαλλήλου. Στον **παράλληλο μονοδιάστατο** πίνακα **tmima_all** να καταχωριστεί το **τμήμα** του/της κάθε υπαλλήλου (**SD** – software department, **MD** – marketing department) και στον **παράλληλο μονοδιάστατο** πίνακα **misthoi_all**, ο **μισθός** του/της.

Τα δεδομένα της εταιρείας «SoftDev» (αρχείο **softdev.txt**) να καταχωρίζονται στις **πρώτες 500** θέσεις και τα δεδομένα της εταιρείας «DigiTask» (αρχείο **digitask.txt**) στις **επόμενες 200** θέσεις των πινάκων.

(Μονάδες 6)

Παράδειγμα:

Ενοποιημένο σύστημα μισθοδοσίας

data_all		tmima_all		misthoi_all		
	0	1				
0	MARIA	GEORGIU	0	SD	0	2856.40
1	GIORGOS	ANDREOU	1	SD	1	2580.00
2	ANDREAS	MICHAEL	2	MD	2	2300.70
⋮	.	.	⋮	.	⋮	.
697	CHRISTOS	MAKRIDIS	697	SD	697	3120.50
698	MARIOS	MARKOU	698	MD	698	1900.00
699	MICHALIS	STYLIANOY	699	MD	699	2150.30

Η **MARIA GEORGIU** εργάζεται στο τμήμα **SD (Software Development)** και ο μισθός της είναι **2856.40** ευρώ.

Στις **πρώτες 500 γραμμές** των πινάκων (**data_all**, **misthoi_all** και **tmima_all**) είναι καταχωρισμένα τα δεδομένα της εταιρείας «SoftDev» (αρχείο **softdev.txt**) και στις **υπόλοιπες 200 γραμμές** είναι αντίστοιχα καταχωρισμένα τα δεδομένα της εταιρείας «DigiTask» (αρχείο **digitask.txt**).

(β) να δέχεται το **όνομα** και το **επίθετο** ενός/μίας υπαλλήλου και να **τυπώνει** στην **οθόνη** το **τμήμα** και τον **μισθό** του/της (με ακρίβεια **δύο (2) δεκαδικών ψηφίων**). Να θεωρήσετε ότι υπάρχει μόνο ένας/μία τέτοιος/τέτοια υπάλληλος.

(Μονάδες 3)

(γ) να χρησιμοποιεί τη **συνάρτηση** με όνομα **insertionsort**, η οποία να δέχεται παραμετρικά τους πίνακες **tmima_all** και **misthoi_all** και να τους **ταξινομεί** σε **φθίνουσα** σειρά με βάση τον μισθό του/της υπαλλήλου, χρησιμοποιώντας τον αλγόριθμο **ταξινόμησης με Εισαγωγή (Insertion sort)**. Ακολούθως, να **εντοπίζει** και να **τυπώνει** στο αρχείο **12out.txt**, από την **κύρια συνάρτηση main**, τους **μισθούς**

(με ακρίβεια **δύο (2) δεκαδικών ψηφίων**) των **πέντε (5) πιο υψηλόμισθων** υπαλλήλων του τμήματος **Software Development**. Η αναζήτηση του τμήματος **Software Development** να γίνεται με το όνομα τμήματος **«SD»**, χρησιμοποιώντας τη μέθοδο της **σειριακής** αναζήτησης. Να θεωρήσετε ότι όλοι οι υπάλληλοι έχουν **διαφορετικό μισθό** και ότι υπάρχουν **τουλάχιστον πέντε (5) υπάλληλοι στο τμήμα SD**.

(Μονάδες 6)

Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων σύμφωνα με το πιο κάτω παράδειγμα:

Παράδειγμα Εισόδου (για 7 υπαλλήλους μόνο από κάθε αρχείο)
(από το αρχείο softdev.txt)

MARIA GEORGIΟΥ SD 2856.40
GIORGOS ANDREOU SD 2580.00
ANDREAS MICHAEL MD 2300.70
PANAYIOTIS PETROY SD 2690.00
NICOLETTA IOANNOU MD 3000.70
NICOS ANASTASIOU MD 1700.25
MARIOS IOANNIDES SD 2457.89

(από το αρχείο digittask.txt)

ANDRIA NEOCLEOUS SD 2300.00
MICHAELLA STAVROY MD 1180.60
CHRISTOS MAKRIDIS SD 3120.50
MARIOS MARKOU MD 1900.00
MICHALIS STYLIANOY MD 2150.30
GEORGIA IACOVIDOU SD 1900.55
ELENA THEOFILOU SD 2367.78

(από το πληκτρολόγιο)

ANDREAS MICHAEL

Παράδειγμα Εξόδου (για τα πιο πάνω δεδομένα)
(στην οθόνη)

Δώσε όνομα και επίθετο του/της υπαλλήλου:

Τμήμα:MD

Μισθός:2300.70

(στο αρχείο 12out.txt)

3120.50

2856.40

2690.00

2580.00

2457.89

ΑΣΚΗΣΗ 12:

Μια εταιρεία που δραστηριοποιείται στον κλάδο του γενικού εμπορίου, επιθυμεί τη μηχανογράφηση των διαδικασιών του τμήματος διαχείρισης των παραγγελιών της. Ο μέγιστος ημερήσιος αριθμός παραγγελιών που μπορεί να διαχειριστεί η εταιρεία είναι **χίλιες (1000) παραγγελίες**.

Στο αρχείο **12in.txt** είναι καταχωρισμένα, για κάθε παραγγελία, ο **αριθμός της παραγγελίας (ακέραιος αριθμός)**, το **όνομα του προϊόντος (συμβολοσειρά)**, η **ποσότητα (ακέραιος αριθμός)**, η **τιμή (δεκαδικός αριθμός)** και ο **κωδικός του πελάτη**

(ακέραιος αριθμός). Ένας πελάτης μπορεί να έχει περισσότερες από μια παραγγελίες και ένα προϊόν μπορεί να παραγγελθεί από περισσότερους από έναν πελάτη.

Να δημιουργήσετε πρόγραμμα στη γλώσσα προγραμματισμού C++ το οποίο:

(α) να ορίζει μια εγγραφή με το όνομα **paragelia**, η οποία να έχει ως μέλη τον αριθμό της παραγγελίας (**ap**), το όνομα του προϊόντος (**pro**), την ποσότητα (**posot**), την τιμή του προϊόντος (**timi**) και τον κωδικό του πελάτη (**pela**). Ακολουθώντας, να ορίζει πίνακα εγγραφών **1000** θέσεων με όνομα **orders**.

(Μονάδες 4)

(β) να διαβάζει τα πιο πάνω στοιχεία από το αρχείο **12in.txt** και να τα καταχωρίζει στον πίνακα εγγραφών **orders**, όπως αυτός έχει οριστεί στο ερώτημα (α). Το πλήθος των παραγγελιών μέσα στο αρχείο είναι άγνωστο και δεν ξεπερνά τις 1000.

(Μονάδες 4)

(γ) να χρησιμοποιεί τη συνάρτηση με όνομα **product**, η οποία να δέχεται παραμετρικά τον πίνακα **orders** και το πλήθος των παραγγελιών και να υπολογίζει και να επιστρέφει στην κύρια συνάρτηση **main** το όνομα του προϊόντος που συνολικά έχει παραγγελθεί τις περισσότερες φορές. Ακολουθώντας, το όνομα του προϊόντος αυτού να τυπώνεται στην οθόνη. Να θεωρήσετε ότι υπάρχει μόνο ένα τέτοιο προϊόν.

(Μονάδες 7)

Το πρόγραμμα πρέπει να εμφανίζει στην οθόνη τα κατάλληλα μηνύματα για την εισαγωγή των δεδομένων και την εξαγωγή των αποτελεσμάτων σύμφωνα με το πιο κάτω παράδειγμα:

Παράδειγμα Εισόδου (από το αρχείο 12in.txt)	(για 13 παραγγελίες μόνο)
124 spaghetti 125 0.45 345 133 sugar 571 0.32 261 111 flour 433 0.24 345 136 lemonade 100 1.15 233 122 bread 235 0.15 261 112 sausages 200 2.15 233 117 ketchup 320 1.60 345 101 rise 145 0.22 250 103 soda 158 0.35 261 147 cheese 170 0.80 233 123 ketchup 180 1.60 250 345 bread 210 0.15 255 134 sugar 340 0.32 250	
Παράδειγμα Εξόδου (στην οθόνη)	(για 13 παραγγελίες μόνο)
Προϊόν με τις περισσότερες παραγγελίες:sugar	

ΤΕΛΟΣ ΕΞΕΤΑΣΤΙΚΟΥ ΔΟΚΙΜΙΟΥ

**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 μια συμβολοσειρά που μπορεί να εισαχθεί από το πληκτρολόγιο ή από αρχείο (x) στο αντικείμενο y.	Η 1 ^η παράμετρος (x) αφορά τη μέθοδο εισαγωγής της συμβολοσειράς (π.χ. από το πληκτρολόγιο ή από αρχείο) και η 2 ^η παράμετρος (y) αφορά το αντικείμενο στο οποίο θα αποθηκευτεί η συμβολοσειρά η οποία έχει διαβαστεί αρχικά.
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 -32767 (στα 2 bytes) ή -2147483648 (στα 4 bytes)	