

**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΑΘΛΗΤΙΣΜΟΥ ΚΑΙ ΝΕΟΛΑΙΑΣ
ΔΙΕΥΘΥΝΣΗ ΜΕΣΗΣ ΕΚΠΑΙΔΕΥΣΗΣ**

**ΕΝΙΑΙΕΣ ΤΕΛΙΚΕΣ ΑΠΟΛΥΤΗΡΙΕΣ ΓΡΑΠΤΕΣ ΕΞΕΤΑΣΕΙΣ 2025 - 2026
Γ΄ ΤΑΞΗΣ ΛΥΚΕΙΟΥ**

ΗΜΕΡΟΜΗΝΙΑ: 20 Μαΐου 2026

ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ:

**ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΕΠΙΣΤΗΜΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Α΄ ΣΕΙΡΑ**

ΚΩΔΙΚΟΣ ΜΑΘΗΜΑΤΟΣ: Γ015

ΣΥΝΟΛΙΚΗ ΔΙΑΡΚΕΙΑ ΓΡΑΠΤΗΣ ΕΞΕΤΑΣΗΣ: 90 λεπτά

ΟΙ ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΛΥΣΕΙΣ ΑΠΟΤΕΛΟΥΝΤΑΙ ΑΠΟ ΔΕΚΑΤΡΕΙΣ (13) ΣΕΛΙΔΕΣ

ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΛΥΣΕΙΣ

ΜΕΡΟΣ Α:**ΑΣΚΗΣΗ 1:**

(α) Δίνεται ο δεκαδικός αριθμός **A=142**. Να δείξετε ότι η αντίστοιχη τιμή του δεκαδικού αριθμού A στο δυαδικό σύστημα είναι **(10001110)₂**, σημειώνοντας τα βήματα που ακολουθήσατε για να φτάσετε στο συγκεκριμένο αποτέλεσμα.

(Μονάδες 2)**Λύση:****A' τρόπος:**

		Πηλίκο	Υπόλοιπο	
Βήμα 1	142/2	71	0	↑
Βήμα 2	71/2	35	1	
Βήμα 3	35/2	17	1	
Βήμα 4	17/2	8	1	
Βήμα 5	8/2	4	0	
Βήμα 6	4/2	2	0	
Βήμα 7	2/2	1	0	
Βήμα 8	1/2	0	1	

B' τρόπος:

2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
128	64	32	16	8	4	2	1
1	0	0	0	1	1	1	0

$$(142)_{10} = (10001110)_2$$

(β) Δίνονται οι δυαδικοί αριθμοί **B=11001010** και **Γ=01010101**. Αφού υπολογίσετε το συμπλήρωμα ως προς 2 του δυαδικού αριθμού Γ, να γράψετε στο δυαδικό σύστημα το αποτέλεσμα της αφαίρεσης **B-Γ**.

(Μονάδες 4)**Λύση:**

$$B=11001010 \text{ και } \Gamma=01010101.$$

Συμπλήρωμα ως προς 2 του Γ: **10101011**

$$\begin{array}{r} B-\Gamma: 11001010+ \\ \quad \underline{10101011} \\ \quad 401110101 \end{array}$$

$$B-\Gamma=(01110101)_2$$

(γ) Δίνεται το πιο κάτω πρόγραμμα στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας οποιονδήποτε τρόπο (π.χ. μέθοδο της προκαταρκτικής εκτέλεσης) να παρουσιάσετε τα αποτελέσματα του προγράμματος. Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```

#include<iostream>
using namespace std;
int process(int &m,int n){
    if (m<n){
        m+=2;
        n-=1;
    }
    else {
        m-=1;
        n+=2;
    }
    cout<<m<<" "<<n<<endl;
    return (m * n) % 5;
}
int main(){
    int result, x = 4, y = 7;
    result = process(y, x);
    cout<<x<<" "<<y<<endl;
    cout<<result<<endl;
    return 0;
}

```

(Μονάδες 5)

Λύση:

Κύρια συνάρτηση(main)

Μεταβλητές			Παρουσίαση
result	x	y	
1	4	7	6 □ 6
		6	4 □ 6
			1

Συνάρτηση process

Τ.Π.Α	Τ.Π.Τ	Απόφαση		Επιστρέφει
		m<n	A/Ψ	
7	4	7<4	Ψ	1
6	6			

(δ) Το πιο κάτω πρόγραμμα, στη γλώσσα προγραμματισμού C++, διαβάζει έναν πραγματικό αριθμό (float) x και υπολογίζει το αποτέλεσμα της πράξης $f=3x+4$. Στη συνέχεια, υπολογίζει και τυπώνει στην οθόνη το δεκαδικό μέρος του αριθμού f, με ακρίβεια 2 δεκαδικών ψηφίων.

Στο πρόγραμμα υπάρχουν **τέσσερα (4) λογικά ή/και συντακτικά** λάθη. Να γράψετε στο τετράδιο απαντήσεών σας, 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<stream>
/*2*/  #include<cmath>
/*3*/  #include<iomanip>
/*4*/  using namespace std;
/*5*/  int main(){
/*6*/      float dec,f,x;
/*7*/      cin>>x;
/*8*/      f=3x+4;
/*9*/      dec = f-round(f);
/*10*/     cout<<fixed<<setw(2)<<dec;
/*11*/     return 0;
/*12*/  }

```

(Μονάδες 4)

Λύση:

```

/*1*/  #include<iostream>
/*8*/  f=3*x+4;
/*9*/  dec=f-trunc(f);
/*10*/ cout<<fixed<<setprecision(2)<<dec;

```

ΑΣΚΗΣΗ 2:

Δίνεται το πιο κάτω πρόγραμμα, στη γλώσσα προγραμματισμού C++, το οποίο διαβάζει από ένα αρχείο τα επίθετα, τις ειδικότητες και τα χρόνια υπηρεσίας καθηγητών/τριών. Από το πρόγραμμα λείπουν μία ή περισσότερες εντολές από τα σημεία A, B, Γ και Δ:

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;



int main(){
    int years, pl=0;
    string surname, eidikotita;
    while(  ){

        filein>>surname>>eidikotita>>years;





    }

```

```
cout<<"Πλήθος καθηγητών/τριών Πληροφορικής ή Φυσικής με
υπηρεσία από 10 έως 20 χρόνια, συμπεριλαμβανομένων: "<< pl;
filein.close();
fileout.close();
return 0;
}
```

(α) Να γράψετε τις κατάλληλες εντολές που απαιτούνται στο σημείο **A** για τη δήλωση μιας ροής ανάγνωσης από το αρχείο **teachers.txt** καθώς και μιας ροής εγγραφής προς το αρχείο **newer.txt**.

(Μονάδες 2)

Λύση:

Θέση A: ifstream filein("teachers.txt");
ofstream fileout("newer.txt");

(β) Να γράψετε τη λογική συνθήκη που απαιτείται στο σημείο **B**, ώστε το πρόγραμμα να συνεχίζει να διαβάζει **μέχρι να φτάσει στο τέλος του αρχείου** teachers.txt.

(Μονάδες 2)

Λύση:

Θέση B: !filein.eof()

(γ) Να γράψετε τις κατάλληλες εντολές που απαιτούνται στο σημείο **Γ**, ώστε το πρόγραμμα να τυπώνει στο αρχείο **newer.txt** τα επίθετα και την ειδικότητα των καθηγητών/τριών των οποίων **τα χρόνια υπηρεσίας είναι λιγότερα από 8**.

(Μονάδες 4)

Λύση:

Θέση Γ: if(years<8)
fileout<<surname<<" "<<eidikotita;

(δ) Να γράψετε τις κατάλληλες εντολές που απαιτούνται στο σημείο **Δ**, ώστε το πρόγραμμα να υπολογίζει το πλήθος των καθηγητών/τριών με ειδικότητα «Πληροφορική» ή «Φυσική», με υπηρεσία **από 10 έως 20 χρόνια**, συμπεριλαμβανομένων.

(Μονάδες 7)

Λύση:

Θέση Δ: if(eidikotita=="Πληροφορική" || eidikotita=="Φυσική")
if(years>=10 && years<=20)
pl++;
ή

if((eidikotita=="Πληροφορική" || eidikotita=="Φυσική") && years>=10 && years<=20)
pl++;

ΑΣΚΗΣΗ 3:

(α) Ένα έξυπνο σύστημα ποτίσματος ελέγχεται από τρεις αισθητήρες: X (υγρασία εδάφους), Y (έξυπνος διακόπτης που ενεργοποιείται εξ' αποστάσεως) και Z (χρονοδιακόπτης/ώρα). Οι καταστάσεις των αισθητήρων είναι οι εξής:

- X: 1 όταν το έδαφος είναι υγρό, 0 όταν είναι στεγνό.
- Y: 1 όταν ο έξυπνος διακόπτης είναι ενεργοποιημένος, 0 όταν είναι απενεργοποιημένος.
- Z: 1 όταν είναι η προγραμματισμένη ώρα ποτίσματος, 0 τις υπόλοιπες ώρες.

Το σύστημα ενεργοποιεί το πότισμα όταν:

- είναι η προγραμματισμένη ώρα ποτίσματος και το έδαφος είναι στεγνό, ή
- είναι ενεργοποιημένος ο έξυπνος διακόπτης

Όταν το πότισμα είναι ενεργοποιημένο, το σύστημα βρίσκεται στην κατάσταση 1.

Να δημιουργήσετε τον **πίνακα αληθείας** για το πιο πάνω αυτοματοποιημένο σύστημα και, ακολούθως, χρησιμοποιώντας αυτόν τον πίνακα, να **γράψετε** την αντίστοιχη **λογική συνάρτηση F** του συστήματος.

(Μονάδες 5)

Λύση:

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$F(X,Y,Z)=X'Y'Z+ X'YZ'+ X'YZ+ XYZ'+ XYZ$$

(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 F(A,B,C,D)** που προκύπτει στην πιο απλή της μορφή.

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

(Μονάδες 6)

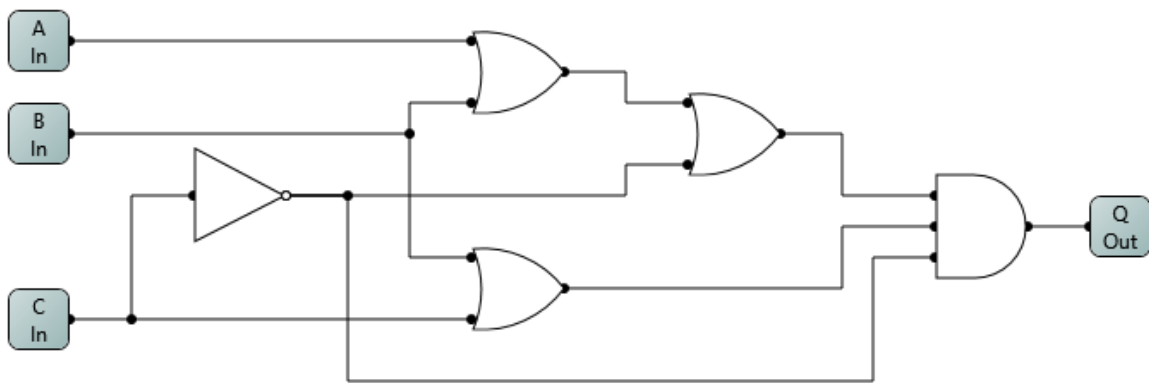
Λύση:

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

Συνάρτηση F στην πιο απλή της μορφή:

$$F(A,B,C,D) = C' + A'D' + BD'$$

(γ) Να γράψετε τη λογική συνάρτηση Q που αντιστοιχεί στο πιο κάτω λογικό κύκλωμα. Στη συνέχεια να βρείτε την τιμή της Q, αν οι μεταβλητές εισόδου παίρνουν τιμές ως ακολούθως: A=0, B=1 και C=0.



(Μονάδες 4)

Λύση:

Λογική συνάρτηση:

$$Q(A,B,C) = (A+B+C') * (B+C) * C'$$

Αποτέλεσμα της συνάρτησης:

$$Q(A,B,C) = (0+1+1) * (1+0) * 1 = 1 * 1 * 1 = 1$$

ΑΣΚΗΣΗ 4:

Ο Γιάννης είναι μελισσοκόμος και έχει τοποθετημένες τις κυψέλες του σε N γραμμές ($3 \leq N \leq 100$). Κάθε γραμμή έχει M κυψέλες ($3 \leq M \leq 100$). Όταν έρθει ο καιρός για να μαζέψει το μέλι από τις κυψέλες, ο Γιάννης ακολουθεί τους πιο κάτω κανόνες:

- i. μαζεύει το μέλι από τις κυψέλες γραμμή προς γραμμή,
- ii. μαζεύει πάντα το μέλι της πρώτης κυψέλης σε κάθε γραμμή,
- iii. μαζεύει το μέλι από μία κυψέλη μόνο αν η προηγούμενη κυψέλη της ίδιας γραμμής δεν έχει καθόλου μέλι.

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

(α) δέχεται το πλήθος N των γραμμών και το πλήθος M των κυψελών, καθώς και την **ποσότητα** μελιού, σε κιλά, που έχει η κάθε κυψέλη, και να την αποθηκεύει στις πρώτες N γραμμές και M στήλες ενός δισδιάστατου πίνακα με το όνομα **posK**.

(Μονάδες 3)

(β) αποθηκεύει στις πρώτες N θέσεις ενός παράλληλου μονοδιάστατου πίνακα με το όνομα **totB**, τα συνολικά κιλά μελιού που θα μαζευτούν από κάθε γραμμή κυψελών, σύμφωνα με τους τρεις κανόνες που ακολουθεί ο Γιάννης. Το πρόγραμμα να τυπώνει στην οθόνη τα περιεχόμενα του πίνακα **totB**.

(Μονάδες 7)

(γ) αποθηκεύει στις πρώτες N θέσεις ενός παράλληλου μονοδιάστατου πίνακα με το όνομα **pliC**, το πλήθος των κυψελών της κάθε γραμμής που έχουν περισσότερα κιλά μέλι από την τελευταία (δεξιότερη) κυψέλη της γραμμής αυτής. Το πρόγραμμα να τυπώνει στην οθόνη τα περιεχόμενα του πίνακα **pliC**.

(Μονάδες 5)

Παράδειγμα Εισόδου	Παράδειγμα Εξόδου
5	Πίνακας totB :
6	7
0 4 5 0 3 0	14
6 4 0 8 6 5	18
3 0 0 9 0 6	5
0 5 6 7 0 0	4
4 4 6 5 6 6	Πίνακας pliC :
	3
	3
	1
	3
	0

Λύση:

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

int main(){
    int N,M,posK[100][100],totB[100],pliC[100];
//α
    cin>>N>>M;
    for( int i=0;i<N; i++ )
        for( int j=0; j<M; j++ )
            cin>>posK[i][j];
//β
    cout<<"Πίνακας totB : "<<endl;
    for( int i=0;i<N; i++ ){
        totB[i]=posK[i][0];
        for( int j=1; j<M; j++ )
            if( posK[i][j-1]==0 )
                totB[i]+=posK[i][j];
        cout<<totB[i]<<endl;
    }
//γ
    cout<<"Πίνακας pliC : "<<endl;
    for( int i=0;i<N; i++ ){
        pliC[i]=0;
        for( int j=0; j<M; j++ )
            if(posK[i][j]>posK[i][M-1])
                pliC[i]++;
        cout<<pliC[i]<<endl;
    }

    return 0;
}
```

ΤΕΛΟΣ Α΄ ΜΕΡΟΥΣ
ΑΚΟΛΟΥΘΕΙ ΤΟ ΜΕΡΟΣ Β΄

ΑΣΚΗΣΗ 5:

Πιο κάτω δίνονται οι βιβλιοθήκες και η κύρια συνάρτηση (main) ενός προγράμματος στη γλώσσα προγραμματισμού C++, το οποίο καλεί διάφορες συναρτήσεις:

```
#include<iostream>
#include<string>
#define N 20
using namespace std;
int main(){
    string lexeis[N];
    float pososto;
    cout<<"Δώσε τις λέξεις του πίνακα:";
    readwords(lexeis);
    swplast(lexeis[0],lexeis[1]);
    cout<<"Λέξεις με αλλαγή γράμματος:"<<lexeis[0]<<" "<<lexeis[1];
    pososto=perc(lexeis);
    cout<<"Ποσοστό εμφάνισης του χαρακτήρα a:"<<pososto<<"%";
    sort(lexeis);
    return 0;
}
```

(α) Να γράψετε τη συνάρτηση **readwords**, η οποία να δέχεται τον πίνακα συμβολοσειρών **lexeis**, να διαβάζει 20 λέξεις και να τις καταχωρίζει στον πίνακα.

(Μονάδες 2)

(β) Να γράψετε τη συνάρτηση **swplast**, η οποία να δέχεται τις δύο συμβολοσειρές που βρίσκονται στις δύο πρώτες θέσεις του πίνακα **lexeis** και να ανταλλάσσει τον τελευταίο χαρακτήρα της μιας συμβολοσειράς με τον τελευταίο χαρακτήρα της άλλης. Στη συνέχεια, να επιστρέφει στο κυρίως πρόγραμμα τις δύο τροποποιημένες συμβολοσειρές.

(Μονάδες 6)

(γ) Να γράψετε τη συνάρτηση **perc**, η οποία να δέχεται τον πίνακα συμβολοσειρών **lexeis** και να υπολογίζει το **ποσοστό** εμφάνισης του χαρακτήρα 'a', στον πίνακα συμβολοσειρών, σε σχέση με το σύνολο των χαρακτήρων του πίνακα. Να θεωρήσετε ότι ο πίνακας περιέχει τουλάχιστον έναν χαρακτήρα 'a'. Στη συνέχεια, η συνάρτηση να επιστρέφει στο κυρίως πρόγραμμα το ποσοστό που έχει υπολογιστεί.

Υπόδειξη: Το ποσοστό υπολογίζεται ως το πηλίκο του πλήθους των χαρακτήρων 'a' προς το συνολικό πλήθος χαρακτήρων του πίνακα, επί 100.

(Μονάδες 6)

(δ) Να γράψετε τη συνάρτηση **sort**, η οποία να δέχεται τον πίνακα συμβολοσειρών **lexeis** και, χρησιμοποιώντας τον αλγόριθμο **ταξινόμησης με εισαγωγή (insertion sort)**, να ταξινομεί σε αύξουσα σειρά **μόνο τις δέκα (10) τελευταίες** συμβολοσειρές του πίνακα. Οι πρώτες δέκα (10) συμβολοσειρές του πίνακα να παραμείνουν στην αρχική τους σειρά.

(Μονάδες 6)

Λύση:

// (α) Συνάρτηση readwords

```
void readwords(string lex[]){
    for(int i=0; i<N; i++)
        cin>>lex[i];
}
```

// (β) Συνάρτηση swaplast

```
void swaplast(string &a, string &b){
    char ch;
    ch=a[a.size()-1];
    a[a.size()-1]=b[b.size()-1];
    b[b.size()-1]=ch;
}
```

// (γ) Συνάρτηση perc

```
float perc(string lex[]){
    int pla=0, plall=0;
    for(int i=0; i<N; i++){
        for(int j=0; j<lex[i].size(); j++){
            plall++;
            if( lex[i][j]=='a' )
                pla++;
        }
    }
    return (float)pla/plall*100;
}
```

// (δ) Συνάρτηση sort

```
void sort(string lex[]){
    int j;
    string temp;
    for(int i=11; i<N; i++){
        temp=lex[i];
        j=i-1;
        while(j>=10 && lex[j]>temp){
            lex[j+1]=lex[j];
            j--;
        }
        lex[j+1]=temp;
    }
}
```

ΑΣΚΗΣΗ 6:

Ένα σχολείο διοργανώνει εκδρομή μαθητών της Α΄ Λυκείου στην Ελλάδα, στην οποία θα συμμετέχουν 80 μαθητές. Να γράψετε πρόγραμμα στη γλώσσα προγραμματισμού C++, το οποίο να:

(α) δημιουργεί μια εγγραφή με το όνομα **excursion**, η οποία να έχει τα πιο κάτω μέλη:

- Επίθετο μαθητή (**surname-string**)
- Τμήμα μαθητή (**tmima-string**)
- Δόσεις (**payments-πίνακας int 5 θέσεων**). Η πρώτη θέση του πίνακα αντιστοιχεί στην πληρωμή της 1^{ης} δόσης, ποσού €100, ενώ οι υπόλοιπες τέσσερις θέσεις αντιστοιχούν σε τέσσερις δόσεις ποσού €200 η καθεμία.

Στη συνέχεια, το πρόγραμμα να **διαβάζει** τα πιο πάνω στοιχεία για όλους τους μαθητές και να τα καταχωρίζει σε έναν πίνακα εγγραφών τύπου **excursion**, με όνομα **excur**. Τα επίθετα των μαθητών δίνονται ταξινομημένα σε αύξουσα αλφαβητική σειρά. Το ποσό για τις δόσεις που δεν έχουν πληρωθεί ακόμη δίνεται ως 0 (μηδέν).

(Μονάδες 5)

(β) **υπολογίζει** και να **τυπώνει** στην οθόνη, το **πλήθος** των μαθητών που πλήρωσαν και τις πέντε (5) δόσεις, δηλαδή έχουν εξοφλήσει το συνολικό ποσό της εκδρομής (€900).

(Μονάδες 3)

(γ) **διαβάζει** το επίθετο ενός μαθητή, ο οποίος, για σοβαρούς προσωπικούς λόγους, δεν θα λάβει τελικά μέρος στην εκδρομή. Να θεωρήσετε ότι υπάρχει μόνο ένας μαθητής με το συγκεκριμένο επίθετο. Στη συνέχεια, να καλεί τη συνάρτηση **binarysearch**, δίνοντάς της τον πίνακα **excur** και το **επίθετο** του μαθητή. Η συνάρτηση, χρησιμοποιώντας **δυναμική αναζήτηση**, να **επιστρέφει** στην κύρια συνάρτηση τον δείκτη της θέσης του μαθητή στον πίνακα **excur**.

(Μονάδες 6)

(δ) **τυπώνει** στην οθόνη τα επίθετα και τα τμήματα όλων των μαθητών που θα λάβουν τελικά μέρος στην εκδρομή, καθώς και το ποσό που οφείλει ακόμη ο κάθε μαθητής. Τα πιο πάνω στοιχεία να τυπώνονται σε ξεχωριστή γραμμή για κάθε μαθητή, με δεξιά στοίχιση 15 στιγμών. Να μην τυπώνονται τα στοιχεία του μαθητή που αποχώρησε από την εκδρομή.

(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 5 μαθητές μόνο)

```
Δημητρίου A21 100 200 200 200 0
Ιωάννου A21 100 200 200 0 0
Κωνσταντίνου A31 100 200 200 200 200
Λεωνίδου A41 100 200 200 200 200
Χρίστου A41 100 200 200 0 0
Ιωάννου
```

Παράδειγμα εξόδου (για 5 μαθητές μόνο)

Δώσε τα στοιχεία για όλους τους μαθητές:

Πλήθος μαθητών που εξόφλησαν : 2

Δώσε επίθετο μαθητή που δεν θα λάβει μέρος:

Δημητρίου	A21	200
Κωνσταντίνου	A31	0
Λεωνίδου	A41	0
Χρίστου	A41	400

Λύση:

```
#include<iostream>
#include<iomanip>
#define N 80
using namespace std;

// (α) - Δημιουργία εγγραφής excursion
struct excursion{
    string surname, tmima;
    int payments[5];
};
// (γ) - συνάρτηση binarysearch για αναζήτηση με βάση το επίθετο με
τη μέθοδο της δυαδικής αναζήτησης.

int binarysearch(excursion excur[],string epitheto){
    int first=0, last=N-1, mid;
    while( first<=last ){
        mid=(first+last)/2;
        if( excur[mid].surname==epitheto)
            return mid;
        else if( epitheto< excur[mid].surname)
            last=mid-1;
        else
            first=mid+1;
    }
}

int main(){
    // (α) - Δήλωση μεταβλητών και καταχώρηση στοιχείων στον πίνακα
    εγγραφών excur όπως περιγράφεται στο ερώτημα (α).
    excursion excur[N];
    int cnt, sum;
    int thesi;
    string epitheto;

    cout<<"Δώσε τα στοιχεία για όλους τους μαθητές :";
    for( int i=0;i<N;i++ ){
        cin>>excur[i].surname>>excur[i].tmima;
        for( int j=0;j<5;j++ )
            cin>>excur[i].payments[j];
    }
}
```

```

// (β) - υπολογισμός πλήθους μαθητών που εξόφλησαν.
cnt=0;
for( int i=0;i<N;i++ ){
    sum=0;
    for( int j=0;j<5;j++ )
        sum+=excur[i].payments[j];
    if(sum==900)
        cnt++;
}
cout<<"Πλήθος μαθητών που εξόφλησαν : "<<cnt<<endl;

// (γ) - κλήση συνάρτησης
cout<<"Δώσε επίθετο μαθητή που δεν θα λάβει μέρος : ";
cin>>epitheto;
thesi = binarysearch( excur, epitheto );

// (δ) - εκτύπωση στοιχείων μαθητών και οφειλόμενου ποσού
for( int i=0;i<N;i++ ){
    if(i!=thesi){
        cout<<setw(15)<<excur[i].surname;
        cout<<setw(15)<<excur[i].tmima;
        sum=0;
        for(int j=0;j<5;j++)
            sum+=excur[i].payments[j];
        cout<<setw(15)<<900-sum<<endl;
    }
}
return 0;
}

```

ΤΕΛΟΣ ΠΡΟΤΕΙΝΟΜΕΝΩΝ ΛΥΣΕΩΝ