

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΑΘΛΗΤΙΣΜΟΥ ΚΑΙ ΝΕΟΛΑΙΑΣ  
ΔΙΕΥΘΥΝΣΗ ΑΝΩΤΕΡΗΣ ΕΚΠΑΙΔΕΥΣΗΣ  
ΥΠΗΡΕΣΙΑ ΕΞΕΤΑΣΕΩΝ

ΠΑΓΚΥΠΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΠΡΟΣΒΑΣΗΣ 2024

ΜΑΘΗΜΑ: ΠΛΗΡΟΦΟΡΙΚΗ (15)

ΕΝΔΕΙΚΤΙΚΕΣ ΑΠΑΝΤΗΣΕΙΣ

ΜΕΡΟΣ Α'

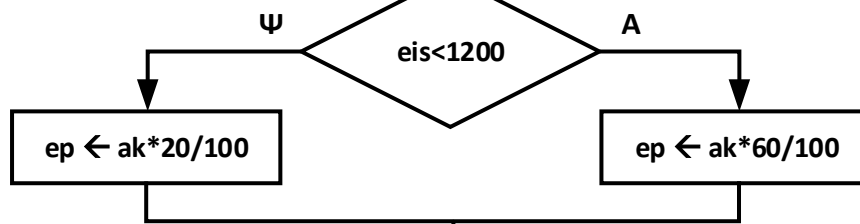
ΑΣΚΗΣΗ 1:

(α)

ΑΡΧΗ

Διάβασε  
eis, ak

(β)



(γ)

Τύπωσε  
tk, ep

ΤΕΛΟΣ

## ΑΣΚΗΣΗ 2:

(α) Μετατροπή του δεκαδικού κλασματικού αριθμού  $A=8.375$  στο δυαδικό σύστημα.

**Ακέραιο Μέρος:**

	Αριθμός	Πηλίκο	Υπόλοιπο
Βήμα 1:	8/2	4	0
Βήμα 2:	4/2	2	0
Βήμα 3:	2/2	1	0
Βήμα 4:	1/2	0	1

$$(8)_{10}=(1000)_2$$

**Κλασματικό Μέρος  
(με 3 ψηφία):**

	Αριθμός	Γινόμενο	Ψηφίο (Bit)
Βήμα 1:	0.375	$0.375 \cdot 2 = 0.75$	0
Βήμα 2:	0.75	$0.75 \cdot 2 = 1.5$	1
Βήμα 3:	0.5	$0.5 \cdot 2 = 1$	1

$$(0.375)_{10}=(011)_2$$

(β) Μετατροπή του δυαδικού αριθμού  $B=(01110001)_2$  στο δεκαδικό σύστημα.

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

$$\begin{aligned} B=(01110001)_2 &= 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 64 + 32 + 16 + 1 \\ &= (113)_{10} \end{aligned}$$

(γ) Το συμπλήρωμα ως προς 2 του  $\Gamma=(00101010)_2$  είναι  $11010110$

0	1	1	1	0	0	0	1
1	1	0	1	0	1	1	0
<del>0</del>	1	0	0	0	1	1	1

Αγνοούμε την υπερχείλιση του τελευταίου ψηφίου

$B-\Gamma = (01000111)_2$

### ΑΣΚΗΣΗ 3:

#### Κύρια συνάρτηση (main)

Μεταβλητές			Παρουσίαση
x	y	z	□□6 □□4
1	2		1 □ 4
	4	5	□□2 □□1
1	1		1 □ 1

#### Συνάρτηση test

ΤΠΤ	ΤΠΑ	ΤΜ	Απόφαση		Επιστρέφει
a	b	c	a>b	T/F	
1	2		1>2	F	5
6	4	5			1
5	1		5>1	T	
2	1	1			

### ΑΣΚΗΣΗ 4:

(α) i) `(num%10!=5)`

ii) `(str1.empty() && str2[str2.size()-1]=='S')`

(β)

i) `x=0 (FALSE)`

ii) `y=1 (TRUE)`

(γ) `/*9*/ };`  
`/*13*/ cava wine[n];`  
`/*17*/ if(wine[i].etos<2010 && wine[i].timi>100){`  
`/*21*/ if(found==false)`

### ΑΣΚΗΣΗ 5:

(α) **Θέση Α:** `ofstream fout("out5.txt", (ios::out|ios::app));`

(β) **Θέση Β:**

```
if (timi[i] > timi[i+1]) {
    temp1=timi[i];
    timi[i]=timi[i+1];
    timi[i+1]=temp1;
    temp2=name[i];
    name[i]=name[i+1];
    name[i+1]=temp2;
    sorted=false;
}
```

(γ) **Θέση Γ:** `fout<<timi[29] - timi[0];`

### ΑΣΚΗΣΗ 6:

```
int search(int nums[], int A, int B){
    int start=0, last=49, mid;
    while(start<=last){
        mid = (start+last)/2;
        if (nums[mid]==B)
            return mid;
        if (nums[mid]==A)
            return mid;
        if (nums[mid]<A)
            start=mid+1;
        else
            last=mid-1;
    }
}
```

## ΜΕΡΟΣ Β'

### ΑΣΚΗΣΗ 7:

(α) Πίνακας αληθείας και λογική συνάρτηση:

$$F(A,B,C,D) = A'BC'D + A'BC'D' + AB'C'D + AB'C'D'$$
$$\text{ή } F(A,B,C,D) = A'BC' + AB'C'$$

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

(β) Λογική συνάρτηση F1:

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

$$F1(A,B,C,D) = A'B' + B'C' + B'D' + C'D'$$

(γ) Λογική συνάρτηση F2:

$$F2(A,B,C,D) = (A'B + C)(A + C')$$

## ΑΣΚΗΣΗ 8:

```
(α) string shiftright(string le){
    char temp=le[le.size()-1];
    for(int i=le.size()-1; i>0; i--)
        le[i]=le[i-1];
    le[0]=temp;
    return le;
}

(β) for(int i=1;i<=20;i++){
    cin>>str;
    stright=shiftright(str);
    cout<<stright<<endl;
}
```

## ΑΣΚΗΣΗ 9:

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

int main(){
    // (α)
    // δήλωση μεταβλητών και ροών ανάγνωσης και εγγραφής στο αρχείο
    ifstream fin("codes.txt");
    ofstream fout("codes.txt");
    int oldk,p2,p3,sumold,sumnew,newk;

    // καταχώρηση στοιχείων
    cout<<"Δώσε παλιό κωδικό και τα δύο μεσαία ψηφία του νέου
κωδικού:"<<endl;

    cin>>oldk>>p2>>p3;
```

```

// (β)
// υπολογισμός όλων των πιθανών τετραψήφιων κωδικών που
// μπορεί να αποτελούν τον νέο κωδικό και καταχώρηση τους στο
// αρχείο codes.txt
sumold=oldk/1000+oldk%1000/100+oldk%100/10+oldk%10;
// ή sumold=oldk/1000+oldk/100%10+oldk/10%10+oldk%10;
for(int i=1000; i<=9999; i++) {
    sumnew= i/1000+i%1000/100+i%100/10+i%10;
    // ή sumnew=i/1000+i/100%10+i/10%10+i%10;
    if (i/100%10==p2 && i/10%10==p3 && sumold - sumnew ==5)
        fout << i << endl;
}
// (γ)
// υπολογισμός και παρουσίαση νέου κωδικού ...
while(!fin.eof()){
    fin>>newk;
    if (newk%10 == oldk/1000)
        cout<<"Νέος κωδικός του πατέρα:"<<newk;
}
fin.close();
fout.close();
return 0;
}

```

### Σημείωση:

Υπάρχουν εναλλακτικές λύσεις π.χ. χειρισμός κωδικού ως συμβολοσειράς, ανάγνωση και διαχείριση κάθε ψηφίου ξεχωριστά και διαχείριση κωδικών των οποίων τα πρώτα ψηφία να είναι μηδέν (π.χ. 0123, 0028, 0009).

## ΑΣΚΗΣΗ 10:

```
#include<iostream>
#include<fstream>
using namespace std;
#define row 101

int main(){
    // (α)
    int elia[row][row], eliarev[row][row];
    int i, j;
    // καταχώρηση στοιχείων στον πίνακα elia
    cout<<"Δώσε το πλήθος των ελαιόδεντρων που υπάρχουν σε
κάθε κτήμα:";
    for (i=0;i<row;i++)
        for (j=0;j<row;j++)
            cin>>elia[i][j];

    // (β)
    // αντιστροφή πίνακα elia και αποθήκευση στον πίνακα eliarev
    for (int i=0; i<row; i++)
        for (int j=0; j<row; j++)
            eliarev[row-1-i][row-1-j] = elia[i][j];
    // τυπώνει τον πίνακα eliarev
    for (i=0;i<row;i++){
        for (j=0;j<row;j++)
            cout<<eliarev[i][j]<<" ";
        cout<<endl;
    }
    // (γ)
    cout << "Δώσε το μέγεθος της πλευράς N:" << endl;
    cin >> N;
    int start = row/2-N/2;
    int finish = row/2+N/2;
    int sum = 0;
    for (int i=start; i<=finish; i++)
        for (int j=start; j<=finish; j++)
            sum += eliarev[i][j];
    sum++;
    cout << "Ελαιόδεντρα που πρέπει να αποκοπούν:" << sum;
    return 0;
}
```



## ΜΕΡΟΣ Γ'

### **ΑΣΚΗΣΗ 11:**

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

#define row 700
#define col 2

// (γ)
// συνάρτηση insertionsort

void insertionsort(string tmima_all[],float misthoi_all[]){
    int i,j;
    float temp_m;
    string temp_t;

    for (i=1;i<row;i++){
        temp_m = misthoi_all[i];
        temp_t = tmima_all[i];
        j=i-1;
        while (j>=0 && misthoi_all[j]<temp_m){
            misthoi_all[j+1] = misthoi_all[j];
            tmima_all[j+1] = tmima_all[j];
            j--;
        }
        misthoi_all[j+1] = temp_m;
        tmima_all[j+1] = temp_t;
    }
}

int main(){

    // (α)
    // δήλωση μεταβλητών

    string data_all[row][col],tmima_all[row],ono,epith;
    int i,j,pl;
    float misthoi_all[row];
    ifstream finso("softdev.txt");
    ifstream findi("digitask.txt");
    ofstream fout("12out.txt");
```

```

// καταχώρηση στοιχείων στους πίνακες

for (i=0;i<500;i++){
    for (j=0;j<col;j++){
        finso>>data_all[i][j];
        finso>>tmima_all[i];
        finso>>misthoi_all[i];
    }
for (i=500;i<700;i++){
    for (j=0;j<col;j++){
        findi>>data_all[i][j];
        findi>>tmima_all[i];
        findi>>misthoi_all[i];
    }
}

// (β)
// εντοπισμός και παρουσίαση των στοιχείων συγκεκριμένου/νης
υπαλλήλου

cout<<"Δώσε όνομα και επίθετο του/της υπαλλήλου: ";
cin>>ono>>epith;
for (i=0;i<row;i++){
    if(data_all[i][0]==ono && data_all[i][1]==epith){
        cout<<"Τμήμα:"<<tmima_all[i]<<endl;
        cout<<"Μισθός:"<<fixed<<setprecision(2)<<misthoi_all[i]<<endl;
    }
}

// (γ)
// εντοπισμός και καταχώρηση στο αρχείο 12out.txt των πέντε πιο
υψηλόμισθων υπαλλήλων του τμήματος Software Development(SD)

insertionsort(tmima_all,misthoi_all);

i=0;
pl=1;
while(i<row && pl<=5){
    if (tmima_all[i]== "SD"){
        fout<<misthoi_all[i]<<endl; // το setprecision(2) δεν
                                // είναι απαραίτητο,
                                // διότι κρατάει τη
                                // ρύθμιση από το μέρος(β)

        pl++;
    }
    i++;
}
return 0;
}

```

## ΑΣΚΗΣΗ 12:

```
#include<iostream>
#include<fstream>
#include<iomanip>
#include<cmath>
using namespace std;

// (α)
// Δημιουργία εγγραφής paragelia. Δήλωση πίνακα orders τύπου
// paragelia

struct paragelia{
    int ap,posot,pela;
    string pro;
    float timi;
} orders[1000];

// (γ)
// συνάρτηση product που υπολογίζει και επιστρέφει το όνομα του
// προϊόντος που συνολικά έχει παραγγελθεί τις περισσότερες φορές.

string product(paragelia ord[], int N){
    int sum=0, max=0;
    string maxp;
    for (int i=0;i<N;i++){
        sum=0;
        for (int j=0;j<N;j++) // σωστό και j=i
            if (ord[i].pro==ord[j].pro){
                sum+=ord[j].posot;
            }
        if (sum>max){
            max=sum;
            maxp=ord[i].pro;
        }
    }
    return maxp;
}
```

### Σημείωση:

Υπάρχουν εναλλακτικές, επίσης αποδεκτές λύσεις.

```

int main(){
// (α)
// Δήλωση μεταβλητών
string maxpro;
ifstream fin ("12in.txt");
int n=0, pe, pl,m=0,codpel;
float sk=0;

// (β)
// καταχώρηση στοιχείων στον πίνακα εγγραφών orders
while (!fin.eof()){
    fin>>orders[n].ap>>orders[n].pro>>orders[n].posot>>
        orders[n].timi>>orders[n].pela;
    n++;
}

// (γ)
maxpro=product(orders,n);
cout<<"Προϊόν που έχει παραγγελθεί τις περισσότερες φορές: "
<<maxpro<<endl;

return 0;
}

```