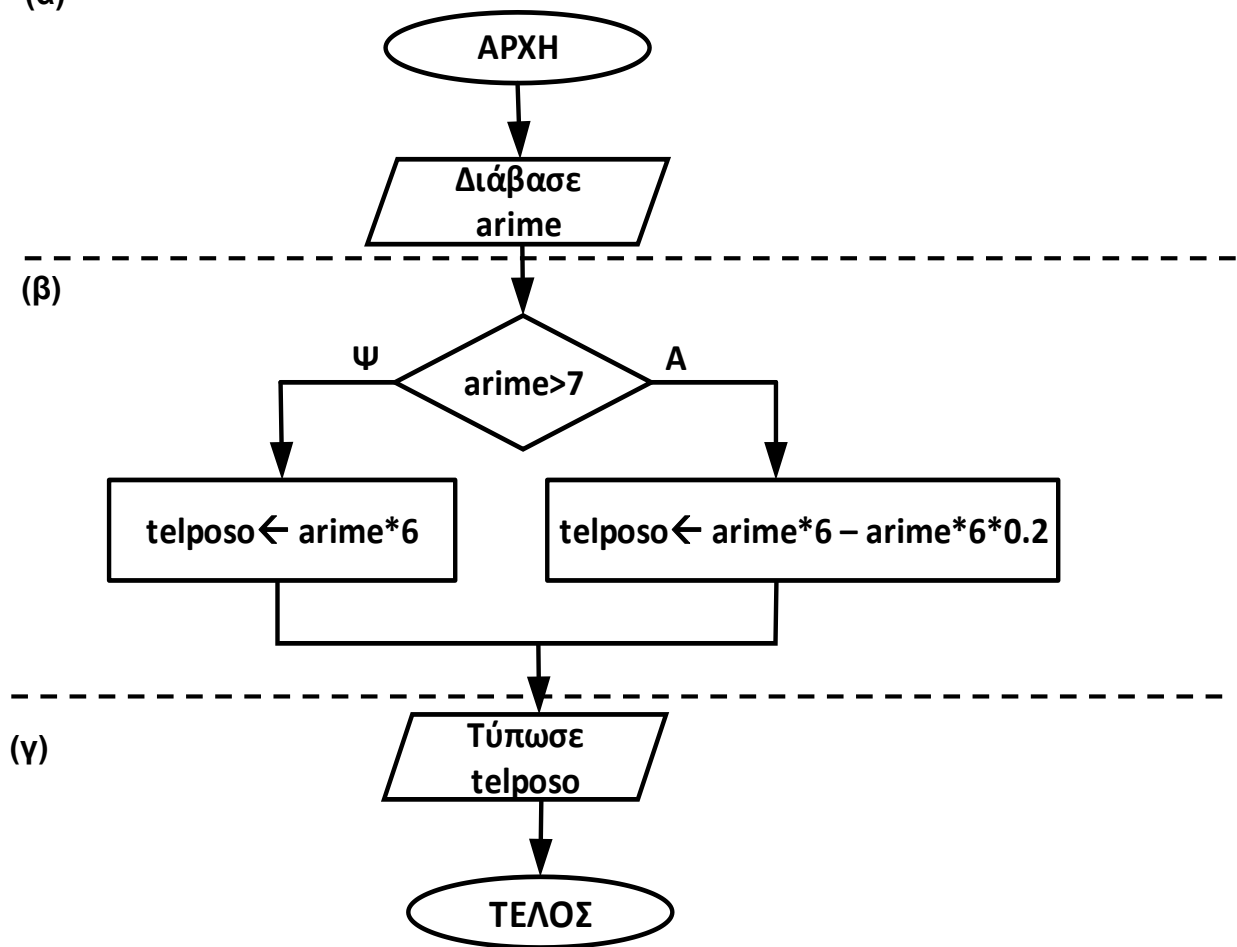


ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΑΘΛΗΤΙΣΜΟΥ ΚΑΙ ΝΕΟΛΑΙΑΣ  
ΔΙΕΥΘΥΝΣΗ ΑΝΩΤΕΡΗΣ ΕΚΠΑΙΔΕΥΣΗΣ  
ΥΠΗΡΕΣΙΑ ΕΞΕΤΑΣΕΩΝ  
ΠΑΓΚΥΠΡΙΕΣ ΕΞΕΤΑΣΕΙΣ ΠΡΟΣΒΑΣΗΣ 2023  
ΠΛΗΡΟΦΟΡΙΚΗ - ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΛΥΣΕΙΣ  
ΜΕΡΟΣ Α'

ΑΣΚΗΣΗ 1:

(α)



## ΑΣΚΗΣΗ 2:

(α) Μετατροπή του δεκαδικού αριθμού  $A=74$  στο δυαδικό σύστημα.

	Αριθμός	Πηλίκo	Υπόλοιπο
Βήμα 1:	74/2	37	0
Βήμα 2:	37/2	18	1
Βήμα 3:	18/2	9	0
Βήμα 4:	9/2	4	1
Βήμα 5:	4/2	2	0
Βήμα 6:	2/2	1	0
Βήμα 7:	1/2	0	1

$$(74)_{10}=(1001010)_2$$

(β) Μετατροπή του δυαδικού αριθμού  $B=(11100010)_2$  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
1	1	1	0	0	0	1	0

$$\begin{aligned} B=(11100010)_2 &= 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 \\ &= 128 + 64 + 32 + 2 \\ &= (226)_{10} \end{aligned}$$

(γ) Το συμπλήρωμα ως προς 2 του  $\Gamma=(01010111)_2$  είναι  $10101001$

1	1	1	0	0	0	1	0	}	$B-\Gamma = (10001011)_2$	
1	0	1	0	1	0	0	1			+
<del>1</del>	0	0	0	1	0	1	1			

Αγνοούμε την υπερχείλιση του τελευταίου ψηφίου

## ΑΣΚΗΣΗ 3:

(α)  $\text{cout} \ll \text{pow}(3, \text{trunc}(3.9)) / \text{sqrt}(10 \% 6 + 5) * \text{abs}(-2 * 5 / 3 + 2);$

$$\begin{aligned} &= \text{pow}(3, 3) / \text{sqrt}(4 + 5) * \text{abs}(-10 / 3 + 2) \\ &= 27 / \text{sqrt}(9) * \text{abs}(-3 + 2) \\ &= 27 / 3 * \text{abs}(-1) \\ &= 9 * 1 \\ &= 9 \end{aligned}$$

(β) i)  $x=1$  (TRUE)  
ii)  $y=1$  (TRUE)

(γ)  $F(A, B, C) = \Sigma(0, 2, 4, 6)$  ή  $F(A, B, C) = m_0 + m_2 + m_4 + m_6$

#### ΑΣΚΗΣΗ 4:

- (α) i) `(num>=100 && num<=999 && num%3==0)`  
ή `(num>99 && num<1000 && num%3==0)`  
ii) `(symv.size())>5 && symv.size()<10)`

```
(β)  switch (code){
        case 22: cout<<"ΛΕΥΚΩΣΙΑ";
                break;
        case 25: cout<<"ΛΕΜΕΣΟΣ";
                break;
        case 24: cout<<"ΛΑΡΝΑΚΑ";
                break;
        case 23: cout<<"ΑΜΜΟΧΩΣΤΟΣ";
                break;
        case 26: cout<<"ΠΑΦΟΣ";
                break;
        default:
                cout<<"Λάθος κωδικός τηλεφώνου";
    }
```

```
(γ)  /*3*/ void foros (float t, float &f, float &tc){
        /*5*/ tc=t + f;
        /*9*/ for (int i=1;i<=5;i++){ ή for (int i=0;i<5;i++){
        /*12*/ foros (timi, fpa, tel_cost);
```

#### ΑΣΚΗΣΗ 5:

- (α) **Θέση Α:** `ifstream fin ("indata.txt");`  
`ofstream fout("outdata.txt", (ios::out|ios::app));`
- (β) **Θέση Β:** `while(!fin.eof())`
- (γ) **Θέση Γ:** `fout<<vath[i]<<endl;`

#### ΑΣΚΗΣΗ 6:

```
(α)  struct car_rent {
        string carnum;
        int etos,plserv;
        string datesrv;
        int kilom[12];
    };
```

(β) `car_rent car;`  
`ή`  
`struct car_rent {`  
 `string carnum;`  
 `int etos,plserv;`  
 `string datesrv;`  
 `int kilom[12];`  
`}car;`

(γ) (i) `cin>>car.carnum>>car.etos;`

(ii) `cout<<car.plserv<<" "<<car.datesrv;`

(iii) `for (i=0;i<12;i++)`  
 `sumkm+=car.kilom[i];`

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

**ΜΕΡΟΣ Β:**

**ΑΣΚΗΣΗ 7:**

(α) Πίνακας αληθείας για την πιο κάτω λογική συνάρτηση:

$$F(A, B, C) = ABC + A'B + A'B'C$$

A	B	C	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	0
1	1	1	1

(β) Λογική συνάρτηση που προκύπτει μετά την ομαδοποίηση των γειτονικών όρων του πιο κάτω χάρτη Karnaugh 4 μεταβλητών στην πιο απλή της μορφή.

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

$$F(A, B, C) = A'BC' + AB'D' + BCD + A'D$$

(γ) Λογική συνάρτηση:

(i)  $F(A, B, C) = A'B'C + AC' + A'B$

Αν  $A=1$ ,  $B=0$  και  $C=1$  ΤΟΤΕ:

(ii)  $F(A, B, C) = A'B'C + AC' + A'B$   
 $= 0+0+0$   
 $= 0$

### ΑΣΚΗΣΗ 8:

```
(α) (i) void maxepisk (int p1, int p2, int p3, int &max){
        max=p1;
        if (p2>max)
            max=p2;
        if (p3>max)
            max=p3;
    }

    (ii) maxepisk(p1,p2,p3,max);
        maxepisk(max,p4,p5,max);
```

(β) Προκαταρκτική εκτέλεση:

Κύρια συνάρτηση (main)

Μεταβλητές		Αποφάσεις		Παρουσίαση
x	y	x<3	A/Ψ	
4	1	-1<3	A	x=-1 □ y=5
6	5	7<3	Ψ	x=7 □ y=9
-1	9			
1				
7				

Συνάρτηση test

Τ.Π.Α.		Αποφάσεις	
a	b	a%2==0	A/Ψ
6	1	6%2==0	A
-1	5	1%2==0	Ψ
1	5		
7	9		

### ΑΣΚΗΣΗ 9:

```
#include<iostream>

#define N 280

using namespace std;
int main (){

    // (α)
    // δήλωση μεταβλητών
```

```

string date[N];
int i, am[N], first=0, last=N-1, mid, cnt=0, target;
bool sorted, found=false;

// καταχώρηση στοιχείων σε αύξουσα σειρά

cout<<"Δώσε αριθμό μητρώου και ημερομηνία γέννησης:"<<endl;
for (i=0; i<N; i++)
    cin>>am[i]>>date[i];

// (β)
// έλεγχος καταχώρησης στοιχείων αν είναι σε αύξουσα σειρά

sorted=true;
for (i=0; i<N-1; i++)
    if (am[i]>am[i+1])
        sorted=false;

// επανάληψη καταχώρησης στοιχείων και έλεγχος μέχρι τα
στοιχεία να καταχωρηθούν σωστά σε αύξουσα σειρά

while (sorted==false){
    cout<<"Επανάληψη καταχώρησης. Αριθμοί μητρώου όχι σε
αύξουσα σειρά!"<<endl;
    for (i=0; i<N; i++)
        cin>>am[i]>>date[i];
    sorted=true;
    for (i=0; i<N-1; i++)
        if (am[i]>am[i+1])
            sorted=false;
}

// (γ)
// διαβάζει τον αριθμό μητρώου που αναζητείται...

cout<<"Δώσε τον αριθμό μητρώου που ψάχνεις:"<<endl;
cin>>target;

// εύρεση αριθμού μητρώου με τη μέθοδο της δυαδικής
αναζήτησης

while (first<=last && found==false){
    mid=(first+last)/2;

```

```

    cnt++;
    // μέτρηση αριθμού επαναλήψεων
    if (target==am[mid])
        found=true;
    else
        if (target<am[mid])
            last=mid-1;
        else
            first=mid+1;
}
// αν ο αρ. μητρώου βρέθηκε παρουσιάζεται η ημερομηνία
γέννησης και αριθμός επαναλήψεων

if (found==true){
    cout<<"Ημερομηνία γέννησης:"<<date[mid]<<endl;
    cout<<"Αριθμός επαναλήψεων:"<<cnt<<endl;
}

// διαφορετικά αν ο αρ. μητρώου δε βρέθηκε τυπώνεται το
κατάλληλο μήνυμα
else
    cout<<"Δεν έχει κάνει εγγραφή!"<<endl;
return 0;
}

```

### ΑΣΚΗΣΗ 10:

```

#include<iostream>
#include<fstream>

using namespace std;

#define row 7
#define col 7

int main(){
    // (α)
    // δήλωση μεταβλητών
    int i,j,scuba[row][col],sum=0,total,max=0;
    // ροή εισόδου fin1 για το αρχείο divers.txt
    ifstream fin1("divers.txt");
    // διαβάζει τα στοιχεία από το αρχείο divers.txt και τα
    καταχωρίζει στον πίνακα scuba
    for (i=0;i<row;i++)
        for (j=0;j<col;j++)
            fin1>>scuba[i][j];
}

```



```

// (β)
// ροή εισόδου fin2 για το αρχείο cord.txt
ifstream fin2("cord.txt");

// υπολογίζει και τυπώνει τον συνολικό αριθμό αντικειμένων
που βρίσκονται στις αντίστοιχες συντεταγμένες
while (!fin2.eof()){ // άγνωστος αριθμός συντεταγμένων
    fin2>>i>>j;
    sum+=scuba[i][j];
}

// παρουσίαση συνολικού αριθμού αντικειμένων
cout<<"Συνολικός αριθμός αντικειμένων:"<<sum<<endl;

// (γ)
// υπολογίζει και τυπώνει τον μέγιστο αριθμό αντικειμένων
που μπορούν να ανασυρθούν από μια περιοχή 2 x 2
for (i=0;i<row-1;i++)
    for (j=0;j<col-1;j++){
        total=scuba[i][j]+scuba[i][j+1]+scuba[i+1][j]+scuba[i+1][j+1];
        if (total>max)
            max=total;
    }

// τυπώνει τον μέγιστο αριθμό αντικειμένων
cout<<"Μέγιστος αριθμός αντικειμένων από μια περιοχή 2x2:"<<max;
fin1.close();
fin2.close();
return 0;
}

```

## ΜΕΡΟΣ Γ:

### ΑΣΚΗΣΗ 11:

```
#include<iostream>
#include <string>

using namespace std;

#define row 100
#define col 10

// (γ)
// συνάρτηση calc

string calc (string names[], char ans[][col]){
    int i,j;
    string newst;
    for (i=0;i<row;i++){
        for (j=0;j<col;j++){
            if (names[i]=="GIORGOS" && ans[i][j]!='N')
                newst=newst+ans[i][j];
        }
    }
    return newst;
}

int main(){

    // (α)
    // δήλωση μεταβλητών

    string names[row],correctans;
    int i,j,vath[row],sum=0;
    char ans[row][col];

    // καταχώρηση στοιχείων στους πίνακες names και ans

    cout<<"Δώσε ονόματα μαθητών/μαθητριών & τις απαντήσεις
τους (A,B,C,D,N):"<<endl;
    for (i=0;i<row;i++){
        cin>>names[i];
        for (j=0;j<col;j++){
            cin>>ans[i][j];
        }
    }
```

```

// (β)
// υπολογισμός συνολικής βαθμολογίας που πέτυχε κάθε
μαθητής / μαθήτρια

```

```

correctans="ABDCBACCCA";
for (i=0;i<row;i++){
    sum=0;
    for (j=0;j<col;j++)
        if (ans[i][j]==correctans[j])
            sum+=5;
        else if (ans[i][j]!='N')
            sum-=2;
    vath[i]=sum;
}

```

```

// (γ)
// παρουσίαση απαντήσεων Γιώργου σε μορφή συμβολοσειράς
χωρίς τα N

```

```

cout<<"Απαντήσεις Γιώργου σε μορφή συμβολοσειράς χωρίς τα
N: "<<calc(names,ans)<<endl;

```

```

return 0;
}

```

## **ΑΣΚΗΣΗ 12:**

```

#include<fstream>

```

```

using namespace std;

```

```

// (α)
// Δημιουργία εγγραφής st και δήλωση πίνακα pro τύπου st

```

```

struct st {
    int am,eisfora;
    string tmima;
} pro[453];

```

```

// (γ)
// συνάρτηση insertionsort για ταξινόμηση σε αύξουσα σειρά με
βάση τον αριθμό μητρώου.

```

```

void insertionsort (st M[], int N){
    int j;
    st temp;
    for (int i=1;i<N;i++){
        temp=M[i];
        j=i-1;
        while (j>=0 && M[j].am>temp.am){
            M[j+1]=M[j];
            j--;
        }
        M[j+1]=temp;
    }
}

int main (){

    // (α)
    // Δήλωση μεταβλητών

    int mitroo, poso, n=0;
    string tm;
    bool flag;

    // (β)
    // καταχώρηση στοιχείων στον πίνακα εγγραφών pro όπως
    // περιγράφεται στο ερώτημα (β)

    ifstream fin ("12in.txt");
    while (!fin.eof()){
        flag=true;
        fin>>mitroo>>poso>>tm;

        for (int i=0;i<n;i++)
            if (pro[i].am==mitroo){
                pro[i].eisfora+=poso;
                flag=false;
            }
        if (flag && poso!=0){
            pro[n].am=mitroo;
            pro[n].eisfora=poso;
            pro[n].tmima=tm;
            n++;
        }
    }
}

```

```
//(γ)
//ταξινόμηση πίνακα pro σε αύξουσα σειρά βάση του αριθμού μητρώου
με τη μέθοδο της ταξινόμησης με εισαγωγή. Ο πίνακας pro τυπώνεται
στο αρχείο 12out.txt

insertionsort(pro,n);

ofstream fout ("12out.txt");
for (int i=0;i<n;i++)
    fout<<pro[i].am<<" "<<pro[i].eisfora<<" "<<pro[i].tmima<<endl;

fin.close();
fout.close();

return 0;
}
```