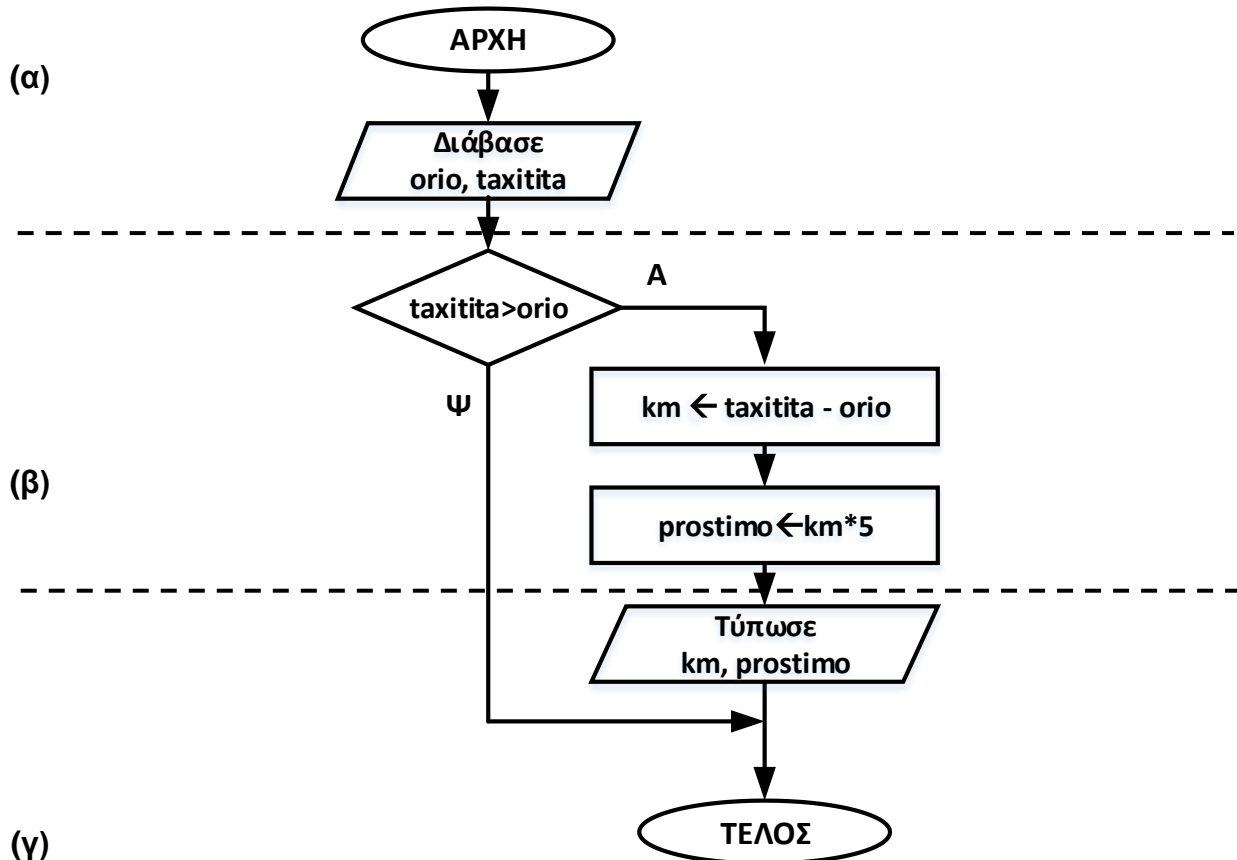


ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΠΟΛΙΤΙΣΜΟΥ
ΔΙΕΥΘΥΝΣΗ ΑΝΩΤΕΡΗΣ ΚΑΙ ΑΝΩΤΑΤΗΣ ΕΚΠΑΙΔΕΥΣΗΣ
ΥΠΗΡΕΣΙΑ ΕΞΕΤΑΣΕΩΝ

ΠΑΓΚΥΠΡΙΕΣ ΕΞΕΤΑΣΕΙΣ 2019
ΠΛΗΡΟΦΟΡΙΚΗ - ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΛΥΣΕΙΣ

ΜΕΡΟΣ Α'

ΑΣΚΗΣΗ 1:



ΑΣΚΗΣΗ 2:

(α) Μετατροπή του δεκαδικού κλασματικού αριθμού **A=3.825** στο δυαδικό σύστημα.

Ακέραιο Μέρος:

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

$(3)_{10} = (11)_2$

Δεκαδικό Μέρος (με 3 δεκαδικά ψηφία):

	Αριθμός	Γινόμενο	Ψηφίο (Bit)
Βήμα 1:	0.825	$0.825 \cdot 2 = 1.65$	1
Βήμα 2:	0.65	$0.65 \cdot 2 = 1.3$	1
Βήμα 3:	0.3	$0.3 \cdot 2 = 0.6$	0

$(0.825)_{10} = (110)_2$

$(3.825)_{10} = (11.110)_2$

(β) Μετατροπή του δυαδικού αριθμού $B=(10001101)_2$ 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	0	0	0	1	1	0	1

$$\begin{aligned}
 B=(10001101)_2 &= 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\
 &= 128 + 8 + 4 + 1 \\
 &= (141)_{10}
 \end{aligned}$$

(γ) Το συμπλήρωμα ως προς 2 του $\Gamma=(01010101)_2$ είναι **10101011**

1	0	0	0	1	1	0	1	
1	0	1	0	1	0	1	1	+
0	0	1	1	1	0	0	0	

Αγνοούμε την υπερχείλιση του τελευταίου ψηφίου

}

$B-\Gamma = (00111000)_2$

ΑΣΚΗΣΗ 3:

(α) `cout<<pow(2,3)+trunc(abs(-8.3))*sqrt(round(63.7));`
 $= 2^3 + \text{trunc}(8.3) * \text{sqrt}(64)$
 $= 8 + 8 * 8$
 $= 8 + 64$
 $= 72$

(β) Αν $a = 5$, $b = 3$ και $c = -1$, τότε:

i) `x = (abs(c) >= b - 4 && c == trunc(sqrt(pow(5,3) - 12)));`
 $x = 0$ (FALSE)

ii) `y = (c + 2*a >= a || !(sqrt(b*8) <= pow(c,4)));`
 $y = 1$ (TRUE)

Σημείωση: Θα ληφθεί υπόψη μόνο το αποτέλεσμα

(γ) $F(A,B,C) = \Sigma(1,3,4,5)$

ΑΣΚΗΣΗ 4:

(α) Οι αντίστοιχες λογικές εκφράσεις για τις πιο κάτω λεκτικές εκφράσεις είναι:

i) Η μεταβλητή **num** να είναι περιττός αριθμός ή μεγαλύτερος του 50

$$(\text{num} \% 2 \neq 0 \ || \ \text{num} > 50) \ \text{ή} \ ((\text{num} \% 2 \neq 0) \ || \ (\text{num} > 50))$$

ii) Η μεταβλητή **plevra** να είναι μεγαλύτερη του 0 και μικρότερη του 10.

$$(\text{plevra} > 0 \ \&\& \ \text{plevra} < 10) \ \text{ή} \ ((\text{plevra} > 0) \ \&\& \ (\text{plevra} < 10))$$

Σημείωση: Τόσο στο i) όσο και στο ii), ένα οποιοδήποτε από τα πιο πάνω θα θεωρηθεί σωστό.

(β) Τα τέσσερα (4) λάθη είναι τα πιο κάτω:

```
/*4*/ #define cols 2 {λάθος 1}
/*6*/ int arr[rows][cols],maxnum,sum=0; {λάθος 2}
/*10*/ cin>>arr[i][j]; {λάθος 3}
/*14*/ if (maxnum<arr[i][j]) {λάθος 4}
```

(γ) Το αντίστοιχο τμήμα προγράμματος με τη χρήση του **while** είναι:

```
k=5;
while (k>0){
    y=k+x;
    cout<<setw(3)<<x<<setw(3)<<y<<endl;
    x+=1;
    k--;
}
```

ΑΣΚΗΣΗ 5:

(α) Θέση Α:

```
ifstream infile("indata.txt");
ofstream outfile("outdata.txt", (ios::out|ios::app));
```

(β) Θέση Β:

```
if (num[i]%2==0)
outfile<<num[i]<<endl;
```

(γ) Θέση Γ:

```
infile.close();
outfile.close();
```

ΑΣΚΗΣΗ 6:

(α)

```
struct date {
    int dd;
    int mm;
    int yyyy;
};

struct student {
    string onoma,address;
    int ar_mit;
    date birth;
};
```

Σημείωση: Η εγγραφή **date** πρέπει να προηγείται της εγγραφής **student**

(β)

```
student mathitis;           ή

struct student {
    string onoma,address;
    int ar_mit;
    date birth;
} mathitis;
```

(γ)

```
cin>>mathitis.onoma;
mathitis.ar_mit=1245;
cout<<mathitis.birth.dd<<mathitis.birth.mm<<mathitis.birth.yyyy;
```

Σημείωση: Η ημερομηνία μπορεί να τυπωθεί σε οποιαδήποτε μορφή.

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

ΜΕΡΟΣ Β'

ΑΣΚΗΣΗ 7:

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

(α) Πίνακας αληθείας της συνάρτησης F.

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

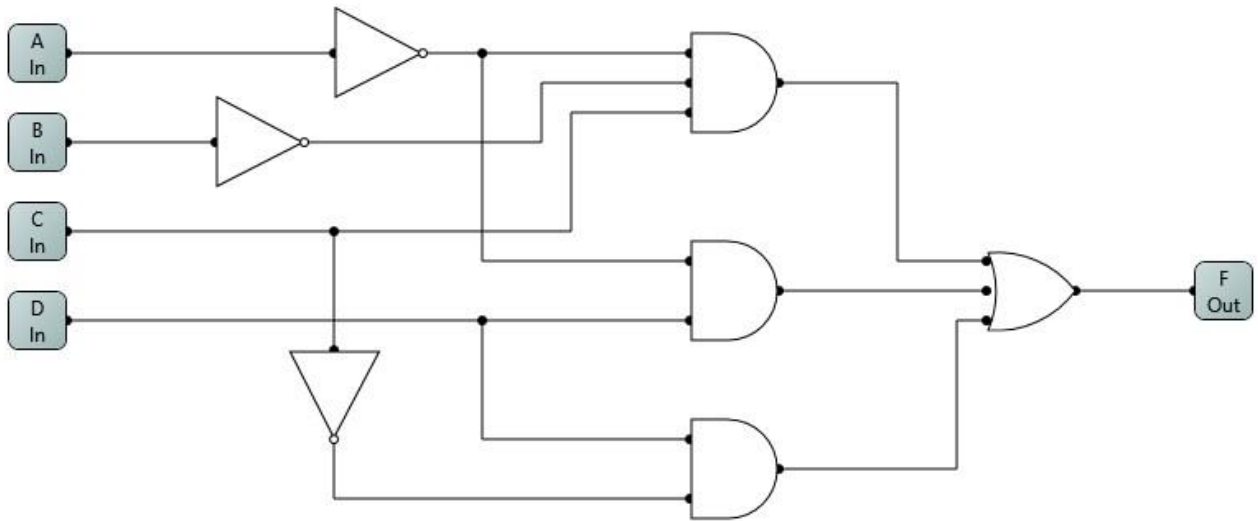
(β) Πίνακας Karnaugh που αντιστοιχεί στη συνάρτηση F.

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

(γ) Απλοποίηση F και αντίστοιχο λογικό κύκλωμα

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

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



ΑΣΚΗΣΗ 8:

(α) (i)

```
void apokopes(float eis, int age, float &fo, float &med){
    fo=0;
    med=0;
    if(age<=65){
        fo=eis*0.2;
        med=eis*0.03;
    }
}
```

(ii)

```
apokopes(eisodima, ilikia, foros, medical);
cout<<"Φόρος:"<<foros<<" "<<"Συνδρομή:"<<medical<<endl;
```

(β) Προκαταρκτική εκτέλεση αν **a=3**, **b=5**, **c=-2** και **d=6**.

Κύρια συνάρτηση (main)

Μεταβλητές				Αποφάσεις		Παρουσίαση
a	b	c	d	c>=10	A/Ψ	
3	5	-2	6	38>=10	A	Δώσε a,b,c και d:
		38	8			c=38 □ d=8

Συνάρτηση calculate

Τ.Π.Τ		Τ.Π.Α	Τοπικές Μεταβλητές	Αποφάσεις	
fa	fb	r1	i	i<=3	A/Ψ
3	5	-2	1	1<=3	A
4	9	38	2	2<=3	A
	13		3	3<=3	A
	17		4	4<=3	Ψ

ΑΣΚΗΣΗ 9:

```
#include<iostream>
#include<string>
using namespace std;
int main(){
    string text,lexi;
    int i,p=0,pchar=0;
    char c1,c2;
    bool find=true;
    cout <<"Διάβασε ένα κείμενο:"<<endl;
    getline (cin, text);
    for (i=0;i<text.size();i++){
        if (text[i]=='.')
            p++;}
    cout<<"Υπάρχουν " <<p<<" προτάσεις"<<endl;
    i=0;
    lexi.clear();
    while (i<text.size() && find==true) {
        if (text[i]!=' ')
            lexi+=text[i];
        else
            find=false;
        i++;
    }
    cout<<"Η 1η λέξη είναι:"<<lexi<<" με μέγεθος " <<lexi.size()<<endl;
    cout<<"Διάβασε δύο χαρακτήρες:";
    cin>>c1>>c2;
    for (i=1;i<text.size();i++)
        if (text[i-1]==c1 && text[i]==c2)
            pchar+=1;
    cout<<"Οι χαρακτήρες " <<c1<<c2<<" εμφανίστηκαν " <<pchar<<" φορές";
    return 0;
}
```

ΑΣΚΗΣΗ 10:

```
#include<iostream>
using namespace std;
#define s 50
int main(){
    int posnum[s],i,temp,minnum;
    bool sorted, found;
```

```

cout<<"Εισαγωγή στοιχείων στον πίνακα:"<<endl;
for (i=0;i<s;i++)
    cin>>posnum[i];
do {
    sorted=true;
    for (i=0;i<s-1;i++) {
        if (posnum[i]>posnum[i+1]){
            temp=posnum[i];
            posnum[i]=posnum[i+1];
            posnum[i+1]=temp;
            sorted=false;
        }
    }
} while (sorted==false);
cout<<"Παρουσίαση ταξινομημένου πίνακα:"<<endl;
for (i=0;i<s;i++)
    cout<<posnum[i]<<" ";

found=false;
if (posnum[0]!=10)
    minnum=10;
else
{
    for (i=0;i<s-1;i++)
        if (posnum[i+1]-posnum[i]!=1 && found==false) {
            minnum=posnum[i]+1;
            found=true;
        }
        if (found==false)
            minnum=posnum[i]+1;
}
cout<<endl<<"Ελάχιστος διψήφιος αριθμός που δεν υπάρχει στον
πίνακα:"<<minnum;
return 0;
}

```

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

ΜΕΡΟΣ Γ'

ΑΣΚΗΣΗ 11:

```
#include<iostream>
#include<iomanip>
using namespace std;
#define row 30
#define col 5
int kamptotal (int fresults[][col],string fwineries[]){
    int i,j,total=0;
    for (i=0;i<row;i++)
        if (fwineries[i]=="MTKampWinery"){
            for (j=0;j<col;j++)
                total+=fresults[i][j];
        }
    return total;
}
int main(){
    string wineries[row];
    int results[row][col],vathmax[row],i,j,maxv,totalMK,sumwine;
    float mo[col];

    cout<<"Δώσε το όνομα κάθε οινοποιείου και τις βαθμολογίες για το κάθε
ένα από τα 5 συγκεκριμένα είδη κρασιών:"<<endl;
    for (i=0;i<row;i++){
        cin>>wineries[i];
        for (j=0;j<col;j++)
            cin>>results[i][j];
    }

    cout<<"Μέγιστη βαθμολογία ανά οινοποιείο:"<<endl;
    for (i=0;i<row;i++) {
        maxv=0;
        for (j=0;j<col;j++)
            if (results[i][j]>maxv)
                maxv=results[i][j];
        vathmax[i]=maxv;
    }
    for (i=0;i<row;i++)
        cout<<vathmax[i]<<" ";
    cout<<endl;
    cout<<"Μέσος όρος βαθμολογίας για το κάθε είδος κρασιού:"<<endl;
```



```

for (j=0;j<col;j++) {
    sumwine=0;
    for (i=0;i<row;i++)
        sumwine+=results[i][j];
    mo[j]= (float) sumwine / row;
}
for (j=0;j<col;j++)
    cout<<mo[j]<<" ";
cout<<endl;
totalMK=kamptotal(results,wineries);
cout<<"Συνολική βαθμολογία του οινοποιείου MTKampWinery:"
<<totalMK<<endl;
return 0;
}

```

ΑΣΚΗΣΗ 12:

```

#include<fstream>
#include<iostream>
#include<cmath>
#include<string>

using namespace std;

#define n 60

struct stathmoi{
    string onoma,eparxia;
    float sixnotitaFM;
    bool pagypria;
};

int main(){
    ifstream fin1("askisi12IN.txt");
    ifstream fin2("akroam.txt");
    ofstream fout1("askisi12aOUT.txt");
    ofstream fout2("askisi12bOUT.txt");
    int i,k,thesi;
    int posakroam[n],diafora, nearest_akroam, tempa, diafafter, diafbefore,
    akrmmradio;
    stathmoi radiost[n],temp;
    string nearest_name;

// διάβασμα στοιχείων ραδιοφωνικών σταθμών από το αρχείο askisi12IN.txt
// διάβασμα ποσοστών ακροαματικότητας από το αρχείο akroam.txt

    for(i=0;i<n;i++){
        fin1>>radiost[i].onoma>>radiost[i].eparxia>>radiost[i].sixnotitaFM>>
        radiost[i].pagypria;
        fin2>>posakroam[i];
    }
}

```

```

// καταχώρηση στοιχείων ραδιοσταθμών που το % ακροαματικότητα είναι > 15
& είναι παγκύπριας εμβέλειας στο αρχείο askisi12aOUT.txt

for(i=0;i<n;i++)
    if (posakroam[i]>15 && radiost[i].pagypria==true)
        fout1<<radiost[i].onoma<<" " <<radiost[i].eparxia<<"
"<<radiost[i].sixnotitaFM<<" " <<radiost[i].pagypria<<"
"<<posakroam[i]<<endl;

// ταξινόμηση των πινάκων radiost και posakroam σε φθίνουσα σειρά (insertion
sort) ως προς το ποσοστό ακροαματικότητας.

for (i=1;i<n;i++){
    temp=radiost[i];
    tempa=posakroam[i];
    k=i-1;
    while (k>=0 && posakroam[k]<tempa) {
        posakroam[k+1]=posakroam[k];
        radiost[k+1]=radiost[k];
        k--;
    }
    posakroam[k+1]= tempa;
    radiost[k+1]=temp;
}

// καταχώρηση στο αρχείο askisi12bOUT.txt των ονομάτων των 5 ραδιοφωνικών
σταθμών με την ψηλότερη συχνότητα εκπομπής

for (i=0;i<5;i++)
    fout2<<radiost[i].onoma<<" " <<posakroam[i]<<endl;

// παρουσίαση του ονόματος και του ποσοστού ακροαματικότητας του
ραδιοφωνικού σταθμού που είναι πιο κοντά (έχει την μικρότερη διαφορά) από
το ποσοστό ακροαματικότητας του ραδιοφωνικού σταθμού MMRADIO.

for(i=0;i<n;i++)
    if (radiost[i].onoma=="MMRADIO") {
        akrmmradio=posakroam[i];
        thesi=i;
    }
    diafbefore=abs(posakroam[thesi-1]-posakroam[thesi]);
    diafafter=abs(posakroam[thesi+1]-posakroam[thesi]);
    if (diafbefore<diafafter){
        nearest_akroam=posakroam[thesi-1];
        nearest_name=radiost[thesi-1].onoma;
    }
    else {
        nearest_akroam=posakroam[thesi+1];
        nearest_name=radiost[thesi+1].onoma;
    }
    cout<<"Όνομα σταθμού:"<<nearest_name<<" " <<"Ποσοστό
ακροαματικότητας:"<<nearest_akroam<<endl;

    fin1.close();
    fin2.close();
    fout1.close();
    fout2.close();

return 0;
}

```

ΤΕΛΟΣ ΕΞΕΤΑΣΤΙΚΟΥ ΔΟΚΙΜΙΟΥ