

**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΠΟΛΙΤΙΣΜΟΥ, ΑΘΛΗΤΙΣΜΟΥ ΚΑΙ ΝΕΟΛΑΙΑΣ
ΔΙΕΥΘΥΝΣΗ ΜΕΣΗΣ ΤΕΧΝΙΚΗΣ ΚΑΙ ΕΠΑΓΓΕΛΜΑΤΙΚΗΣ ΕΚΠΑΙΔΕΥΣΗΣ
ΚΑΙ ΚΑΤΑΡΤΙΣΗΣ**

ΕΝΙΑΙΑ ΓΡΑΠΤΗ ΑΞΙΟΛΟΓΗΣΗ ΤΕΤΡΑΜΗΝΩΝ 20 20 - 20 21

Β' ΤΑΞΗΣ ΤΕΣΕΚ

ΣΕΙΡΑ Α'

ΗΜΕΡΟΜΗΝΙΑ : Πέμπτη, 3 Ιουνίου 2021

ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ: Εφαρμογές Προγραμματισμού II -TEM2

ΚΩΔΙΚΟΣ ΜΑΘΗΜΑΤΟΣ : thips202

ΛΥΣΕΙΣ

ΜΕΡΟΣ Α΄: Αποτελείται από δέκα (10) ερωτήσεις. Κάθε ορθή απάντηση βαθμολογείται με πέντε (5) μονάδες.

1. Εξηγήστε τη διαφορά μεταξύ των τοπικών μεταβλητών (local variables) και των καθολικών μεταβλητών (global variables) στον προγραμματισμό.

Οι τοπικές μεταβλητές μπορούν να χρησιμοποιηθούν μέσα σε μία δηλωμένη συνάρτηση, όμως δεν είναι προσβάσιμες από την κύρια συνάρτηση (main). Αντίθετα, μία καθολική μεταβλητή η οποία δηλώνεται εκτός της κύριας συνάρτησης, μπορεί να είναι προσβάσιμη από την κύρια συνάρτηση αλλά και από όσες συναρτήσεις ορίζονται σε αυτήν.

2. Να συμπληρώσετε όλες τις απαραίτητες εντολές για την πιο κάτω συνάρτηση, ή οποία υπολογίζει το άθροισμα τριών ακέραιων αριθμών.

```
#include <iostream>
using namespace std;
int add_3_nums( int a, int b, int c )
{
return a + b + c;
}
int main() {
cin>>x;
cin>>y;
cin>>z;
cout << add_3_nums( x, y, z);
return 0;}
```

3. Να γράψετε τα αποτελέσματα των πιο κάτω λογικών εκφράσεων οι οποίες είναι γραμμένες στη γλώσσα προγραμματισμού C++, αν $a = 1$, $b = 2$ και $c = -1$.

(i) `cout << ((b>a) && !(a==b));`

true η 1

(ii) `cout << (((b-c)==0) || (b%2==0) && ((b+c)==3));`

false η 0

4. Η χρήση συναρτήσεων και η δομημένη ανάπτυξη προγραμμάτων παρουσιάζει αρκετά πλεονεκτήματα. Να αναφέρετε και να επεξηγήσετε δύο (2) από αυτά τα πλεονεκτήματα.

Διευκολύνεται η αποσφαλμάτωση

Μειώνεται η έκταση του κώδικα

Διασφαλίζεται η σωστή και απλούστερη δημιουργία προγραμμάτων

Διευκολύνεται η κατανόηση του τρόπου λειτουργίας ενός προγράμματος

5. Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++.

```
#include<iostream>
#include<string>
using namespace std;
int main( ) {
string st1, st2;
st2.clear( );
cin >> st1;
for ( int i=0; i<st1.size( ); i++ ) {
    if ( st1[ i ]>='A' && st1[ i ]<='Z' )
        st2 += st1[i]; }
if (st2.empty( ) )
    cout << "No capital letters found";
else
    cout << st2;
return 0;}
```

Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος όταν δοθούν ως είσοδος οι πιο κάτω συμβολοσειρές:

- (i) IMPortPROgram

IMPPRO

- (ii) import PROGRAM

No capital letters found

6. Να αναφέρετε τι είναι οι πίνακες στη γλώσσα προγραμματισμού C++.

Στη C++, όπως και σε όλες τις γλώσσες προγραμματισμού, υπάρχει η δυνατότητα ομαδοποίησης δεδομένων ίδιου τύπου. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό είναι με τη δημιουργία πινάκων (arrays). Αναφορά στον πίνακα γίνεται μέσω ενός ονόματος(αναγνωριστικού), όπως στις μεταβλητές, ενώ για πρόσβαση στα δεδομένα ενός πίνακα χρησιμοποιείται το όνομα και ένας αριθμός (δείκτης - index), ο οποίος καθορίζει τη θέση του στοιχείου μέσα στον πίνακα.

ή

Ο πίνακας είναι μία συλλογή δεδομένων του ίδιου τύπου, τα οποία είναι αποθηκευμένα σε διαδοχικές θέσεις μνήμης.

ή

Ο πίνακας είναι μια σύνθετη μεταβλητή που καταλαμβάνει παραπάνω από μια θέση στην μνήμη του Η/Υ, έχει ένα συγκεκριμένο όνομα και δέχεται ένα συγκεκριμένο τύπο δεδομένων.

7. Να συμπληρώσετε το πιο κάτω πρόγραμμα στη γλώσσα προγραμματισμού C++, το οποίο όταν εκτελεστεί υπολογίζει τον μέσο όρο των ακέραιων αριθμών που υπάρχουν μέσα στο αρχείο "IN.txt" και καταχωρεί το αποτέλεσμα στο αρχείο "OUT.txt".

```
#include<iostream>
#include< fstream>
using namespace std;
#define n 50
int main( ) {
ifstream inf("IN.txt");
ofstream outf ("OUT.txt");
int i, A [ n ], sum=0;
float avg;
for ( i=0; i<n; i++ ) {
inf>>A [ i ];
sum+=A [ i ];
}
avg=sum/n;
outf<<avg<<endl;
inf.close();
outf.close()
return 0;}
```

8. Δίνονται οι πιο κάτω πίνακες:

$\text{int } A[5] = \{10, 20, 30, 40, 50\}, B[5] = \{10, 5, 4, 2, 8\}, C[5] = \{0, 0, 0, 0, 0\};$

Να βρείτε τα αποτελέσματα των πιο κάτω εντολών:

- (i) $C[0] = A[0] * B[0];$ **100**
- (ii) $C[1] = A[3] - B[1];$ **35**
- (iii) $C[2] = A[3] / B[2];$ **10**
- (iv) $C[3] = A[1] - B[0] + B[3] * A[2];$ **20-10+60 = 70**
- (v) $C[4] = A[4]\%B[4];$ **2**

9. Σημειώστε ποιες από τις πιο κάτω δηλώσεις της C++ είναι **ΟΡΘΕΣ** και ποιες είναι **ΛΑΘΑΣΜΕΝΕΣ**.

(i) `int primes[] = {1,2,3,5,7};` **ΟΡΘΟ**

(ii) `double Nums[];` **ΛΑΘΟΣ**

(iii) `char characters[3] = {'a', 'b', 'c', 'd'};` **ΛΑΘΟΣ**

(iv) `int a[5] = {12,5,4};` **ΟΡΘΟ**

(v) `bool logic[4] = {16, 18, 15, 20};` **ΛΑΘΟΣ**

10. Με τη χρήση προκαταρκτικής εκτέλεσης, να παρουσιάσετε τα αποτελέσματα του πιο κάτω προγράμματος.

```
#include <iostream>
using namespace std;
int main()
{
    int A[4]={10,20,30,40};
    int B[4]={0,10,20,30};
    for (int i=0; i<4; i++)
        cout << "X" << i << " = " << A[i] + B[i] << endl;
    return 0;
}
```

ΑΠΑΝΤΗΣΗ

```
X0 = 10
X1 = 30
X2 = 50
X3 = 70
```

ΜΕΡΟΣ Β΄: Αποτελείται από πέντε (5) ερωτήσεις. Κάθε ορθή απάντηση βαθμολογείται με οκτώ (8) μονάδες.

11. Να ξαναγράψετε το πιο κάτω πρόγραμμα, χρησιμοποιώντας πρώτα την δομή **while** και στην συνέχεια την δομή **for**.

<pre>#include<iostream> using namespace std; int main(){ int x, sum = 0, i = 0, num; cin >> x; do { cin >> num; sum += num; i++; } while(i < x); cout << sum; return 0; }</pre>	<p style="text-align: center;"><u>ΑΠΑΝΤΗΣΗ με while</u></p> <pre>#include <iostream> using namespace std; int main () { int x, sum=0 , i=0, num; cin >> x; while(i<x) { cin>>num; sum+=num; i++; } cout << sum; return 0; }</pre>
<pre>#include<iostream> using namespace std; int main(){ int x, sum = 0, i = 0, num; cin >> x; do { cin >> num; sum += num; i++; } while(i < x); cout << sum; return 0; }</pre>	<p style="text-align: center;"><u>ΑΠΑΝΤΗΣΗ με for</u></p> <pre>#include <iostream> using namespace std; int main (){ int x, sum=0 , i=0, num; cin >> x; for(i=0; i<x; i++) { cin>>num; sum+=num; } cout << sum; return 0; }</pre>

12. Το πιο κάτω πρόγραμμα δέχεται δέκα ακέραιους αριθμούς και τυπώνει τον μεγαλύτερο αριθμό καθώς επίσης και τη σειρά στην οποία δόθηκε. Σε περίπτωση που υπάρχουν περισσότεροι από έναν μέγιστοι αριθμοί, τυπώνεται ο τελευταίος που δόθηκε. Στο πρόγραμμα υπάρχουν τέσσερα (4) λογικά ή/και συντακτικά λάθη. Να εντοπίσετε και να διορθώσετε τα λάθη αυτά στην αντίστοιχη γραμμή. Στο πρόγραμμα να μη γίνει καμία προσθήκη ή αφαίρεση εντολής.

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     int num, max, maxth;
5.     cin>>num;
6.     max=num;
7.     for (int i=1; i<=10; i++)
8.         cin>>num;
9.     if(max>num){
10.        maxth=num;
11.        max=num;}
12.     cout << "Max= " <<max<<" thesi= " <<maxth;
13.     return 0;
14. }
```

Γραμμή 7. Εφόσον ο πρώτος αριθμός διαβάστηκε εκτός βρόχου επανάληψης πρέπει να διαβαστούν άλλοι εννέα όχι 10.

Γραμμή 7. for (int i=1; i<=10; i++) {

Γραμμή 9. if(num>max){

Γραμμή 10. maxth=i;

Γραμμή 11 max=num;} }

13. Να γράψετε ξανά το πιο κάτω ολοκληρωμένο πρόγραμμα, χρησιμοποιώντας μια συνάρτηση με όνομα q_par και παράμετρο ένα ακέραιο αριθμό q που θα αντικαθιστά τις μεταβλητές y_par, z_par και f_par.

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

int main()
```

```

{
    int y, z, n, f, y_par=1, z_par=1, f_par=1;
    cin>>y>>z>>f;
    for (int i=1; i<=y; i++)
        y_par *=i;
    for (int j=1; j<=z; j++)
        z_par *=j;
    for (int n=1; n<=f; n++)
        f_par *=n;
    int x=y_par+z_par-f_par;
    cout << x << endl;
    return 0;
}

```

```

#include<iostream>
using namespace std;

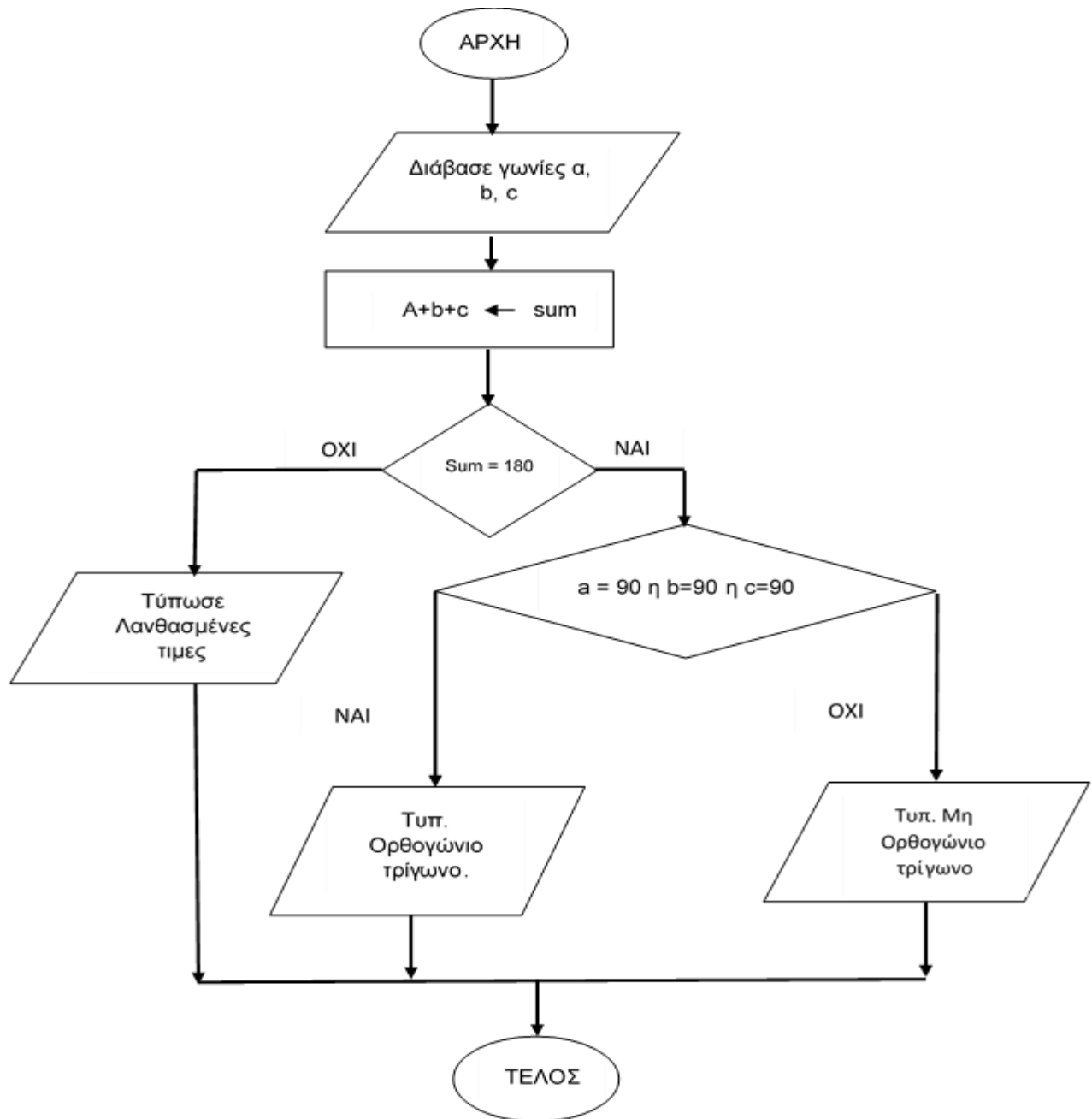
```

```

int q_par(int q){
int parag=1;
for(int i=1; i<=q; i++)
    parag*=i;
return parag;
}
int main(){
int y, z, f;
cin>>y>>z>>f;
int x=q_par(y)+q_par(z)-q_par(f);
cout << x << endl;
return 0;}

```


14. Να μετατρέψετε το πιο κάτω λογικό διάγραμμα σε πρόγραμμα της C++.



```
#include <iostream>
using namespace std;
int main()
{
int a, b, c, sum;
cin>>a>>b>>c;
sum=a+b+c;
if (sum==180)
    if (a==90 || b==90 || c==90)
        cout<<"Orthogonio trigono"<<endl;
```

```

else cout<<"Mi Orthogonio trigono"<<endl;
else cout<<"Lanthasmenes times"<<endl;
return 0;
}

```

15. Να γράψετε πρόγραμμα με την χρήση της δομής επιλογής **switch** που να διαβάζει δύο αριθμούς από τον χρήστη και στη συνέχεια να εμφανίζει στην οθόνη το παρακάτω **menu** επιλογών:

<pre> ----- Επιλέξτε πράξη ----- 1. Πρόσθεση 2. Αφαίρεση 3. Πολλαπλασιασμός 4. Διαίρεση 5. Έξοδος ----- Επιλογή: </pre>

Ανάλογα με την επιλογή να εκτελεί την αντίστοιχη πράξη (διασφαλίζοντας ότι δεν θα γίνει η διαίρεση εάν ο διαιρέτης είναι 0). Στο τέλος να εμφανίζει στην οθόνη το αποτέλεσμα και την πράξη που εκτελέστηκε.

```

#include<iostream>
#include<iomanip>
#include<cstdlib>
using namespace std;
int main()
{
double a, b;
cin>>a>>b;
cout<<"-----"<<endl;
cout<<"Epilexte Praxi"<<endl;
cout<<"-----"<<endl;
cout<<"1. Prothesi"<<endl;
cout<<"2. Afairesi "<<endl;
cout<<"3. pollaplasiasmos"<<endl;
cout<<"4. Diairesi"<<endl;
cout<<"5. Exodos"<<endl;
cout<<"-----"<<endl;
char praxi;

```

```
cout<<"Epilogi:";
cin>>praxi;
switch (praxi){
case '1':
cout<<a+b<<endl;
break;
case '2':
cout<<a-b<<endl;
break;
case '3':
cout<<a*b<<endl;
break;
case '4':
if (b!=0)
cout<<a/b<<endl;
cout<<"Lathos timi diaireti"<<endl;
break;
default:
    break;
}
return 0;
}

return 0;
}
```

ΜΕΡΟΣ Γ': Αποτελείται από μία (1) ερώτηση. Η ορθή απάντηση βαθμολογείται με δέκα (10) μονάδες.

16. Το Υπουργείο Υγείας ανακοινώνει καθημερινά τα αποτελέσματα από τα τεστ ταχείας ανίχνευσης αντιγόνου (rapid test) όπως το παράδειγμα πιο κάτω:

A/A	Επαρχία/Τομέας	Αριθμός Τεστ	Αριθμός Θετικών
1	Λευκωσία	22959	27
2	Λεμεσό	14239	44
3	Λάρνακα	8293	30
4	Πάφος	5820	5
5	Αμμόχωστος	3106	10
6	Εθνική Φρουρά	161	0
7	Οίκοι Ευγηρίας	231	1
8	Αθλητές	181	1

Ο αριθμός των επαρχιών/τομέων κάθε μέρα είναι διαφορετικός.

Το Υπουργείο Υγείας σας ζητά να γράψετε ένα πρόγραμμα στην C++ για να βοηθήσετε να βρεθούν οι πιο κάτω πληροφορίες:

Το κυρίως πρόγραμμα **δίνεται σε εσάς** στο τέλος του εξεταστικού δοκιμίου αυτού και θα καλεί τις ακόλουθες συναρτήσεις με την σειρά:

- i. ReadData
- ii. DisplayData
- iii. MaxPositive
- iv. SaveData

i. Η συνάρτηση ReadData θα ζητά από τον χρήστη το πλήθος των επαρχιών/τομέων και ακολούθως θα ζητά για κάθε επαρχία/τομέα το όνομα, Αριθμός Τεστ και Αριθμός Θετικών όπου θα καταχωρούνται σε τρεις παράλληλους πίνακες (arrays), AreaSector, TestNums και PositiveTest αντιστοίχως.

[3 μονάδες]

iii. Η συνάρτηση DisplayData θα πρέπει να υπολογίζει αρχικά για κάθε επαρχία/τομέα το Ποσοστό Θετικότητας ($= \text{Αριθμός Θετικών} / \text{Αριθμός Τεστ} * 100$). Θα υπολογίζει επίσης το συνολικό Αριθμό Τεστ, το συνολικό Αριθμό Θετικών Τεστ και το ποσοστό θετικότητας επί του συνόλου των τεστ. Τέλος θα εμφανίζει όλους τους πίνακες παράλληλα όπως το παράδειγμα πιο κάτω.

[3 μονάδες]

A/A	Επαρχία/Τομέας	Αριθμός Τεστ	Αριθμός Θετικών	Ποσοστό Θετικότητας
1	Λευκωσία	22959	27	0.12%
2	Λεμεσό	14239	44	0.31%
3	Λάρνακα	8293	30	0.36%
4	Πάφος	5820	5	0.09%
5	Αμμόχωστος	3106	10	0.32%
6	Εθνική Φρουρά	161	0	0.00%
7	Οίκοι Ευγηρίας	231	1	0.43%
8	Αθλητές	181	1	0.55%
	Σύνολο	54990	118	0.21%

- iv. Ο κώδικας στην συνάρτηση MaxPositive θα πρέπει να εμφανίζει επαρχία/τομέα με τον μέγιστο αριθμό θετικών τεστ. Στη περίπτωση όπου δύο οι περισσότερες επαρχίες τομείς έχουν τον ίδιο μέγιστο αριθμό τότε να εμφανίζονται όλες οι περιπτώσεις.

[2 μονάδες]

- v. Ο κώδικας για στην τελευταία συνάρτηση SaveData θα πρέπει να αποθηκεύει την επαρχία/τομέα, Αριθμό Τεστ και τον Αριθμό Θετικών σε ένα αρχείο με όνομα Tests.txt. Η συνάρτηση αυτή θα πρέπει να κάνει έλεγχο εάν το αρχείο έχει ανοίξει και είναι έτοιμο να δεχθεί δεδομένα και να εμφανίζει το ανάλογο μήνυμα.

[2 μονάδες]

Σημείωση:

Όλες οι μεταβλητές και οι πίνακες (arrays) δεν θα πρέπει να δηλωθούν ως καθολικοί (global) αλλά σαν τοπικοί (local) και να περνάνε στις συναρτήσεις με τις παραμέτρους. Καθολικά (global) μπορούν και θα πρέπει να δηλωθούν μόνο σταθερές, πχ τα μεγέθη των πινάκων (Array Size).

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <fstream>
```

```
using namespace std;
```

```
const int Arr_Size = 20;
```

```
//
```

```
void ReadData (string RAreaSector[Arr_Size], int RTestNums[Arr_Size],int  
RPositiveTest[Arr_Size], int & Rplithos)
```

```
{
```

```
    int nCount;
```

```
    cout << "Dose to plithos ton eparhion/tomeon: ";
```

```
    cin >> Rplithos;
```

```
    for (nCount = 0 ; nCount < Rplithos; nCount++)
```

```
    {
```

```
        cout<< "Dose tin eparhia/Tomea ["<<nCount+1<<"]: ";
```

```
        cin >> RAreaSector[nCount];
```

```
        cout<< "Dose ton arithmo ton tests stin eparhia/Tomea
```

```
"<<RAreaSector[nCount]<<": ";
```

```

    cin >> RTestNums[nCount];
    cout<< "Dose ton arithmo ton thetikon tests stin eparhia/Tomea
"<<RAreaSector[nCount]<<": ";
    cin >> RPositiveTest[nCount];
}
cout<<endl;
}
// _____

void DisplayData(string DAreaSector[Arr_Size], int DTestNums[Arr_Size],int
DPositiveTest[Arr_Size], int DPlithos)
{
    int nCount;

    cout<<fixed<<setprecision(2);

    cout<<setw(4)<<"A/A" <<setw(18)<<"Eparxia/Tomeas" <<setw(15)<<"Arithmos
tests";
    cout<<setw(25)<<"Arithmos thetikon tests" <<setw(22)<<"Pososto
Thetikotitas" <<endl;
    for (nCount = 0 ; nCount < DPlithos; nCount++)
    {
        cout<< setw(4)<<nCount+1<<". " <<setw(15)<<DAreaSector[nCount];
        cout<<setw(12)<<DTestNums[nCount]<<setw(20)<< DPositiveTest[nCount];
        cout<<setw(20)<<float(DPositiveTest[nCount])/DTestNums[nCount]*100<<"%" <<
endl;
    }
    cout<<endl;
}
// _____

int MaxPositive (string AreaSector[Arr_Size], int PositiveTest[Arr_Size], int
Mplithos)
{
    int nCount, MaxNum ;

    MaxNum = PositiveTest[0];

```

```

for (nCount = 1 ; nCount < Mplithos; nCount++)
{
    if (MaxNum < PositiveTest[nCount])
        MaxNum = PositiveTest[nCount];
}
for (nCount = 0 ; nCount < Mplithos; nCount++)
{
    if (MaxNum == PositiveTest[nCount])
    {
        cout<<setw(18)<<"Eparxia/Tomeas"<<setw(14)<< AreaSector[nCount];
        cout<<setw(25)<<"Arithmos thetikon
tests"<<setw(5)<<PositiveTest[nCount]<<endl;
    }
}
return MaxNum;
}
// _____
void SaveData(string SAreaSector[Arr_Size], int STestNums[Arr_Size],int
SPositiveTest[Arr_Size], int SPLithos)
{
    int nCount;

    cout<<fixed<<setprecision(2);
    ofstream TestsFile ("Tests.txt");
    if (TestsFile.is_open())
    {

        for (nCount = 0 ; nCount < SPLithos; nCount++)
        {

TestsFile<<setw(15)<<SAreaSector[nCount]<<setw(12)<<STestNums[nCount]<<set
w(20)<<SPositiveTest[nCount]<<endl;
        }
        cout << "Successfully saved" <<endl;
        TestsFile.close();
    }
}

```

```
else
    cout << "Unable to open the file..." <<endl;
    cout<<endl;
}

//_____

int main ()
{
    int Plithos, TestNums[Arr_Size], PositiveTest[Arr_Size];
    string AreaSector[Arr_Size];

    ReadData (AreaSector, TestNums, PositiveTest, Plithos);
    DisplayData (AreaSector, TestNums, PositiveTest, Plithos);
    MaxPositive (AreaSector, PositiveTest, Plithos);
    SaveData (AreaSector, TestNums, PositiveTest,Plithos);

    return 0;}

```