



Υποεργασία 1

A. `a=6` //variable a starts at 6.

```
printf("%d\n",++a);
```

`/* a is first augmented by 1 , then evaluated by printf # stdout reads 7 , a is 7*/`

```
printf("%d\n",a--);
```

`/* a is evaluated by printf, then diminished by 1 # stdout reads 7 , a is 6*/`

```
printf("%d\n",a<<2);
```

`/* a = $6_{10} = 110_2$ # bits are shifted 2 positions to the left,result in $11000_2 = 24$ # a evaluated by printf # stdout reads 24 , a is still 6*/`

```
printf("%d\n",a>>1);
```

`/*# a's bits are shifted 1 positions to the right, result in $11_2 = 3$ # a evaluated by printf # stdout reads 3 , a is still 6*/`

```
printf("%d\n",a^a);
```

`/* bitwise XOR operation between 110_2 and 110_2 # $110_2 \text{ XOR } 110_2 = 000_2$ # stdout reads 0 , a is still 6 */`

B.

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int n;
```

```
    printf("Give the number of days\n");
```

```
    scanf("%d", &n);
```

```
    printf("%d year(s), %d month(s), %d day(s)", n/360,(n  
    %360)/30, (n%360)%30);
```

```
    return (0);
```



}

```
Give the number of days
782
2 year(s), 2 month(s), 2 day(s)

-----
(program exited with code: 0)
Press return to continue
```

Γ.

Calling **func(4);**

void func(int k)

```
{
    if (k==0)
        return;
    printf("%d", k-1);
    int i;
    for (i=1; i<k; i++)
        func(i);
}
```

/*

func(4) → **prints 3** →

i=1 → func(1) → **prints 0** #for loop evaluates false - func(1) terminates.

i=2 → func(2) → **prints 1** # for loop: i=1 → func(1) → **prints 0** #for loop evaluates false - func(2) terminates.

i=3 → func(3) → **prints 2** # for loop: i=1 → func(1) → **prints 0**
func (2) → **prints 1** # for loop: i=1 → func(1) → **prints 0**
#for loop evaluates false - func(2) terminates.

*/

```
30102010

-----
(program exited with code: 4)
Press return to continue
```



Δ. Λόγω των μεγεθών και ελλείπει σχετικής διευκρίνησης, περιοριζόμαστε στην χρήση μεταβλητών τύπου integer.

```
#define EMPLOYEE 20    // No of employees as constant.

/*function that accepts annual income as integer and returns
amount withheld as annual tax. */

int computeTax(int value) {
    int tax;
    if (value<=25000)
        tax=value*0.22;
    else if (value<=42000)
        tax=25000*0.22+(value-25000)*0.32;
    else
        tax=25000*0.22+42000*0.32+(value-42000)*0.42;
    return tax;
}

i.
/* declare counter, temp, and arrays holding employee gross &
net incomes*/
int i, a, grossInc[EMPLOYEE], netInc[EMPLOYEE] ;

/* User to input annual income for each employee, stored in
the grossInc array - def. prog. for negative values */
for (i = 0; i < EMPLOYEE; i++)
{
    printf("Please enter Annual gross income of employee
    No. %d:\n",i+1);
    scanf("%d",&a);
    while (a<0){
        printf("Negative values are not accepted.
        Please try again.\n");
        scanf("%d",&a);
    }
}
```



```
grossInc[i] = a;
```

```
}
```

ii.

```
/* Prints annual tax withheld from salary of each employee,  
using the function computeTax */
```

```
for (i = 0; i < EMPLOYEE; i++)
```

```
{
```

```
printf("Annual Tax amount withheld from employee No. %d  
is: EUR %d\n", i+1, computeTax(grossInc[i]));
```

```
}
```

iii.

```
/* Prints net monthly salary for each employee */
```

```
for (i = 0; i < EMPLOYEE; i++)
```

```
{
```

```
/*populate the netInc array , then divide net income by  
12 to get monthly salary */
```

```
netInc[i] = grossInc[i] - computeTax(grossInc[i]);
```

```
printf("Net monthly salary earned by employee No. %d is:  
EUR %.d\n", i+1, netInc[i]/12 );
```

```
}
```



iv.

```
/* Finds and prints the highest net annual income , along with
its associated tax */

    int max , counter ;
    max = counter =0;
    for (i = 0; i < EMPLOYEE; i++)
    {

        if (max < netInc[i] )
        {
            max = netInc[i];
            counter = i;
        }
    }

    printf("Highest annual Net Income earned by employee No.
%d , \nin the amount of EUR %d , after EUR %d of tax were
withheld.",counter+1,max, computeTax(grossInc[counter]));
```

```
Please enter Annual gross income of employee No. 1:
12000
Please enter Annual gross income of employee No. 2:
25000
Please enter Annual gross income of employee No. 3:
50000
Please enter Annual gross income of employee No. 4:
75000
Annual Tax amount withheld from employee No. 1 is: EUR 2640
Annual Tax amount withheld from employee No. 2 is: EUR 5500
Annual Tax amount withheld from employee No. 3 is: EUR 22300
Annual Tax amount withheld from employee No. 4 is: EUR 32800
Net monthly salary earned by employee No. 1 is: EUR 780
Net monthly salary earned by employee No. 2 is: EUR 1625
Net monthly salary earned by employee No. 3 is: EUR 2308
Net monthly salary earned by employee No. 4 is: EUR 3516
Highest annual Net Income earned by employee No. 4 ,
in the amount of EUR 42200 , after EUR 32800 of tax were withheld.

-----
(program exited with code: 0)
Press return to continue
```



Υποεργασία 2

A.

```
#define N 8      // πλήθος N των κριτηρίων
#define M 3      //αριθμό M των χωρών

const char *criterion[] = {"Κυβερνητική
σταθερότητα","Σταθερότητα εθνικού νομίσματος","Φορολογική
πολιτική","Εγγύτητα σε αγορές","Εγγύτητα σε
προμηθευτές","Μορφωτικό επίπεδο εργαζομένων","Κουλτούρα,
νοοτροπία, παραγωγικότητα","Κόστος εργασίας"} ;    //περιγραφές
των κριτηρίων.

int weights[N];      // συντελεστές βαρύτητας κάθε κριτηρίου.
int scores[N][M];    // βαθμολογία κάθε χώρας ανά κριτήριο.
int finalScores[M];  // τελικό βαθμό για κάθε χώρα.
```



B.

```
void read_weights()
{
    int i,sum =0;
    for (i = 0; i < N; i++)
    {
        printf("Συντελεστής Βαρύτητας για το κριτήριο %d (%s):",i+1,criterion[i] );
        scanf("%d",&weights[i]);
        while (weights[i]<0 || weights[i]>100) //αμυντικός
                                                    προγραμματισμός
        {
            printf("Δεν επιτρέπονται αρνητικές τιμές.
            Ξαναπροσπαθήστε\n");
            scanf("%d",&weights[i]);
        }
        sum+=weights[i];
        printf("SUM(Wj)= %d\n",sum);
    }
    while (sum !=100) //αμυντικός προγραμματισμός
    {
        printf("Το άθροισμα των συντελεστών πρέπει είναι
        100. Επανεκκίνηση ρουτίνας.\n");
        read_weights();
    }
}
```



Γ.

```
void read_scores()
{
    int i,j;
    for (j = 0; j < M; j++)    // iterate through countries
    {
        characters(20,'-');    // add delimiter line
        for (i = 0; i < N; i++) // iterate through criteria
        {
            printf("Βαθμός στο κριτήριο %d για τη χώρα %d:
            ",i+1, j+1);
            scanf("%d",&scores[i][j]);
            while (scores[i][j]<0 || scores[i][j]>5)
            //αμυντικός προγραμματισμός
            {
                printf("Τιμή εκτός ορίου [0,5].
                Ξαναπροσπαθήστε\n");
                scanf("%d",&scores[i][j]);
            }
        }
    }
}
```




Δ.

```
void best_country()
{
    int i,j,cScore, maxScore,index;
    maxScore = index = 0;          //initialize variables
    for (j = 0; j < M; j++)        // iterate through countries
    {
        cScore = 0;
        for (i = 0; i < N; i++)    // iterate through criteria
        {
            cScore += weights[i]*scores[i][j];
            // use formula given to compute score
        }
        finalScores[j] = cScore ;  // add to array
        printf("Τελικός βαθμός για τη χώρα %d:\t\n",j+1,cScore);
    }
    // if score is the highest one yet, set maxScore to that value
    if ( maxScore < cScore) maxScore = cScore;
}
characters(20,'-');              // add delimiter line
printf("Η/Οι χώρα/ες με την υψηλότερη βαθμολογία είναι\n");
for (i = 0; i < M; i++)          // iterate through countries
{
    if (finalScores[i] == maxScore) printf("%d\n",i+1); // find highest score and print its index(es).
}
}
```



Ακολουθεί ένα παράδειγμα εκτέλεσης για $N=5$ και $M=2$:

```
Συντελεστής Βαρύτητας για το κριτήριο 1 (Κυβερνητική σταθερότητα): 12
SUM(Wj)= 12
Συντελεστής Βαρύτητας για το κριτήριο 2 (Σταθερότητα εθνικού νομίσματος): 23
SUM(Wj)= 35
Συντελεστής Βαρύτητας για το κριτήριο 3 (Φορολογική πολιτική): 34
SUM(Wj)= 69
Συντελεστής Βαρύτητας για το κριτήριο 4 (Εγγύτητα σε αγορές): 23
SUM(Wj)= 92
Συντελεστής Βαρύτητας για το κριτήριο 5 (Εγγύτητα σε προμηθευτές): 8
SUM(Wj)= 100
-----
Βαθμός στο κριτήριο 1 για τη χώρα 1: 22
Τιμή εκτός ορίου [0,5].Ξαναπροσπαθήστε
2
Βαθμός στο κριτήριο 2 για τη χώρα 1: 3
Βαθμός στο κριτήριο 3 για τη χώρα 1: 2
Βαθμός στο κριτήριο 4 για τη χώρα 1: 1
Βαθμός στο κριτήριο 5 για τη χώρα 1: 5
-----
Βαθμός στο κριτήριο 1 για τη χώρα 2: 2
Βαθμός στο κριτήριο 2 για τη χώρα 2: 2
Βαθμός στο κριτήριο 3 για τη χώρα 2: 3
Βαθμός στο κριτήριο 4 για τη χώρα 2: 1
Βαθμός στο κριτήριο 5 για τη χώρα 2: 4
-----
Τελικός βαθμός για τη χώρα 1:      224
Τελικός βαθμός για τη χώρα 2:      227
-----
Η/Οι χώρα/ες με την υψηλότερη βαθμολογία είναι η/οι :2
-----
(program exited with code: 0)
Press return to continue
```



Υποεργασία 3

Όπου ήταν δυνατό, έγινε προσπάθεια να χρησιμοποιηθεί η ειδική σημειολογία στην αναφορά του δείκτη s στο σώμα των συναρτήσεων. Στην προκειμένη περίπτωση, η κλήση των στοιχείων του πίνακα συμβατικά (ex. $s[i]$) αποφέρει το ίδιο αποτέλεσμα.

A.

/* συνάρτηση dyn, δέχεται ως ορίσματα μια διπλή x και μια ακέραια k ; επιστρέφει το x^k . */

```
double dyn(double x,int k)
{
    int i;
    double z =1;
    for (i = 0; i < k; i++) z *=x;
    return z;
}
```

B.

/*συνάρτηση p, δέχεται ως ορίσματα δείκτη s σε πίνακα που περιέχει συντελεστές πολυωνύμου, διπλή x , και ακέραια n που αντιστοιχεί στο βαθμό πολυωνύμου; επιστρέφει την τιμή του πολυωνύμου συμβατικά*/

```
double p(double *s, double x, int n)
{
    int i;
    double z =0;
    for (i = 0; i <=n ; i++)
    {
        z+= *(s+i)*dyn(x,i);
    }
    return z;
}
```

Γ.

/*συνάρτηση h, δέχεται ως ορίσματα δείκτη s σε πίνακα που περιέχει συντελεστές πολυωνύμου, διπλή x , και ακέραια n που αντιστοιχεί στο βαθμό πολυωνύμου; επιστρέφει την τιμή του πολυωνύμου κατά Horner */

```
double h(double *s, double x, int n)
{
    int i;
    double z =*(s+n);    //initialize with high. deg.
                           coefficient
    for (i = n; i >0 ; i--)    //reverse iteration;
    {
        z= z*x + *(s+(i-1)) ;
    }
    return z;
}
```



Δ.

```
int main(void)
{
//i. // declare coefficient array & variable x
    double c[10], x;
    // declare polynomial degree
    int i,b;
//ii.
    printf("Enter the polynomial degree: ");
    scanf("%d",&b);
    while (b<0 || b>10)    //αμυντικός προγραμματισμός
    {
        printf("Degree must be positive and less than 10.
            Try again : ");
        scanf("%d",&b);
    }
//iii.
    printf("Enter the value of the constant term: ");
    scanf("%lf",&c[0]);
    for (i = 1; i <=b ; i++)
    {
        printf("Enter the value of the %d degree
            coefficient: ", i);
        scanf("%lf",&c[i]);
    }
    printf("Enter the value of variable x: ");
    scanf("%lf",&x);
//iv.
    printf("The value of the polynomial using the
    conventional method is : %.3lf\n",p(c,x,b));

    printf("The value of the polynomial using the Hohner
    method is : %.3lf\n",h(c,x,b));

    return (0);
```



}

Υποεργασία 4

Χωρίς διευκρίνηση από την εκφώνηση της άσκησης και έχοντας ως οδηγό την εικόνα του πρότυπου προγράμματος στην εκφώνηση, επιλέξαμε να δηλώσουμε τις μεταβλητές N και $size$ ως σταθερές λόγω ευκολίας. Λόγω της χρήσης των μεταβλητών στον ορισμό πινάκων στις μεταβλητές εισόδου των συναρτήσεων, η μόνη απαίτηση για τις N , $size$ είναι να έχουν καθολική εμβέλεια.

A.

```
/* global variables */
const int size=4;
const int N=3;

/* functions */
void read_frequencies(int F[][N], int Q[], int size)
{
    int i,j;
    // Populate the F array
    for (i = 0; i < size; i++)
    {
        for (j = 0; j < N; j++)
        {
            printf("Δώσε την συχνότητα εμφάνισης του όρου
            t%d στο κείμενο d%d:",i,j);
            scanf("%d",&F[i][j]);
            while (F[i][j] < 0)
            {
                printf("Δεν επιτρέπονται μη αρνητικές τιμές.
                Εαναπροσπαθήστε :");
                scanf("%d",&F[i][j]);
            }
        }
    }
}
```



```
// Populate the Q array
for (i = 0; i < size; i++)
{
    printf("Δώσε την συχνότητα εμφάνισης του όρου t%d
στο ερώτημα q: ", i);
    scanf("%d",&Q[i]);
    while (Q[i] < 0)
    {
        printf("Δεν επιτρέπονται μη αρνητικές τιμές.
Εαναπροσπαθήστε :");
        scanf("%d",&Q[i]);
    }
}
}
```

B.

```
void invert_frequencies(int F[][N], double idf[], int size)
{
    int i, j;
    double freq;

    //iterate through tokens
    for (i = 0; i < size; i++)
    {
        freq = 0.0;
        //iterate through texts
        for (j = 0; j < N; j++)
        {
            // record occurrences for the token in each text
            if (F[i][j]>0) freq +=1.0;
        }
        // compute the inverse frequency for the token
        idf[i] = size / freq ;
    }
}
```



```
}
```

Γ.

```
void report_query(int F[][N], int Q[], double idf[], double
similarity[N], int size)

{

    int i,j;

    double wText[size][N], wQuery[size], numeratorSum,
    wTxtSqr, wQrSqr;

    //initialize function variables

    numeratorSum = wTxtSqr = wQrSqr = 0;

    for (j = 0; j < N; j++)

        {

            for (i = 0; i < size; i++)

                {

                    //calculate text weight per token

                    wText[i][j] = F[i][j]*idf[i];

                    //calculate query weight per token

                    wQuery[i] = Q[i]*idf[i];

                    // calculate formula numerator and the
                    series contained in the denominator

                    numeratorSum += wText[i][j] * wQuery[i];

                    wTxtSqr += pow(wText[i][j], 2.0);
```



```
wQrSqr += pow(wQuery[i], 2.0);

}

// apply the formula

similarity[j] = numeratorSum /
( sqrt(wTxtSqr) * sqrt(wQrSqr) );

printf("Κείμενο d%d:  %lf\n", j,
similarity[j]);

// reset function variables to zero for next
iteration

numeratorSum = wTxtSqr = wQrSqr = 0;

}

}
```

```
Δώσε την συχνότητα εμφάνισης του όρου t0 στο κείμενο d0:0
Δώσε την συχνότητα εμφάνισης του όρου t0 στο κείμενο d1:0
Δώσε την συχνότητα εμφάνισης του όρου t0 στο κείμενο d2:1
Δώσε την συχνότητα εμφάνισης του όρου t1 στο κείμενο d0:0
Δώσε την συχνότητα εμφάνισης του όρου t1 στο κείμενο d1:2
Δώσε την συχνότητα εμφάνισης του όρου t1 στο κείμενο d2:0
Δώσε την συχνότητα εμφάνισης του όρου t2 στο κείμενο d0:1
Δώσε την συχνότητα εμφάνισης του όρου t2 στο κείμενο d1:0
Δώσε την συχνότητα εμφάνισης του όρου t2 στο κείμενο d2:1
Δώσε την συχνότητα εμφάνισης του όρου t3 στο κείμενο d0:0
Δώσε την συχνότητα εμφάνισης του όρου t3 στο κείμενο d1:1
Δώσε την συχνότητα εμφάνισης του όρου t3 στο κείμενο d2:0
Δώσε την συχνότητα εμφάνισης του όρου t0 στο ερώτημα q: 0
Δώσε την συχνότητα εμφάνισης του όρου t1 στο ερώτημα q: 0
Δώσε την συχνότητα εμφάνισης του όρου t2 στο ερώτημα q: 2
Δώσε την συχνότητα εμφάνισης του όρου t3 στο ερώτημα q: 0

Ομοιότητες:
Κείμενο d0:  1.000000
Κείμενο d1:  0.000000
Κείμενο d2:  0.447214

-----
(program exited with code: 0)
Press return to continue
```