

**Θέμα 1 (15%):** (απαιτούμενος χρόνος < 15 λεπτά)

Εκτελέστε ‘με το χέρι’ το παρακάτω πρόγραμμα και γράψτε όλες τις ενδιάμεσες τιμές και τις τιμές που τυπώνονται:

```
#include <stdio.h>

int y;
int f(int x, int z);

main()
{
    y = 0;
    printf("%3d ", f(3,2));
    printf("%3d\n", y);
}

int f(int x, int z)
{
    printf("%3d %3d %3d\n", x, y, z);
    if (x < 1)
        z=10;
    else
        y=f(x-1, 1+z)-z;
    printf("%3d %3d %3d\n", x, y, z);
    return (y+z+x);
}
```

**Απάντηση:**

y
0
6
8
11

f(3,2):

x	z
3	2

f(2,3):

x	z
2	3

f(1,4):

x	z
1	4

f(0,5):

x	z
0	5

Εκτυπώσεις:		
3	0	2
2	0	3
1	0	4
0	0	5
0	0	10
1	6	4
2	8	3
3	11	2
16	11	

**Θέμα 2 (15%):** (απαιτούμενος χρόνος < 15 λεπτά)

Να γράψετε ένα πρόγραμμα που να αποφασίζει αν κάποιο έτος είναι δίσεκτο. Το πρόγραμμά σας θα πρέπει:

- να δέχεται έναν αριθμό **year** ανάμεσα στο **1** και το **2500**,
- να υπολογίζει αν το **year** είναι δίσεκτο έτος ή όχι και να εκτυπώνει το αντίστοιχο αποτέλεσμα στην οθόνη.

Ένα έτος είναι δίσεκτο αν διαιρείται με το **4** αλλά όχι με το **100** ή αν διαιρείται με το **400**.

**Απάντηση:**

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

```
main()
```

```
{
```

```
    int year;
```

```
    printf("Dwse moy to etos [1 - 2500]\n");
```

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

```
    if ((year >=1) && (year <= 2500))
```

```
    {
```

```
        printf("To etos %d ",year);
```

```
        if (((year % 4)==0) && ((year % 100)!=0) || ((year % 400)==0))
```

```
            printf("einai disekto\n");
```

```
        else
```

```
            printf("den einai disekto\n");
```

```
    }
```

```
    else
```

```
        printf("Lathos dedomena\n");
```

```
}
```

**Θέμα 3 (25%):** (απαιτούμενος χρόνος < 25 λεπτά)

Δίνεται ο παρακάτω αναδρομικός τύπος υπολογισμού των όρων μιας ακολουθίας:

$$S_0=0, S_1=2, S_{n+2}=2S_n-S_{n+1}$$

Να φτιάξετε ένα πρόγραμμα για τον υπολογισμό και την εκτύπωση των **n** πρώτων όρων της ακολουθίας. Το πρόγραμμά σας θα πρέπει να διαβάζει το **n** από το πληκτρολόγιο και να εκτυπώνει τους **n** πρώτους όρους στην οθόνη. Επίσης το πρόγραμμα θα πρέπει να είναι 'φίλικό' προς τον χρήστη, δηλαδή να επαναλαμβάνεται η εκτέλεση μέχρι ο χρήστης να δώσει την τιμή **-1**.

**Απάντηση:**

```
#include <stdio.h>
main()
{
    int k,n;
    int s0, s1, sn;

    printf("Dwse moy ton ari8mo tw n orwn (-1 gia telos)\n");
    scanf("%d",&n);
    while (n!= -1)
    {
        s0=0;
        printf("Oroi akoloy8ias: %d ",s0);
        s1=2;
        printf("%d ",s1);

        /* Se ka8e epanalhch ypologizetai o oros k san to a8roisma tw n dyo
        prohgoymenwn (k-1), (k-2) poy einai apo8hkeymenoi stoys s1 kai s0
        antistoixa. Akoloy8ws, o oros k apo8hkeyetai sto s1 kai o oros k-1
        apo8hkeyetai sto s0, etsi wste na xrhsimopoih8oyn gia ton ypologismo toy
        oroy k+1 sthn epomenh epanalhch */

        for (k=2;k<n;k++)
        {
            sn=2*s0-s1;
            s0=s1;
            s1=sn;
            printf("%d ",sn);
        }
        printf("\n");
        scanf("%d",&n);
    }
}
```

**Θέμα 4 (45%):** (απαιτούμενος χρόνος < 45 λεπτά)

Να γράψετε ένα πρόγραμμα που να καταχωρεί ακέραιους αριθμούς σε έναν πίνακα **N** θέσεων (**N** μια σταθερά του προγράμματος) ως εξής: κάθε φορά που δίνεται ένας ακέραιος αριθμός **x**,

- αν ο **x** υπάρχει στον πίνακα να διαγράφεται η πρώτη του εμφάνιση,
- αν ο **x** δεν υπάρχει στον πίνακα, να εισάγεται στην πρώτη θέση του πίνακα.

Το πρόγραμμα θα δέχεται **N** ακέραιους από το χρήστη και θα τους εισάγει με τον παραπάνω τρόπο στον πίνακα. Στο τέλος θα πρέπει να τυπώνονται όλα τα στοιχεία του πίνακα.

**Προσοχή:** Τα στοιχεία που θα έχει τελικά ο πίνακας μπορεί να είναι λιγότερα από **N**.

**Απάντηση:**

```
#include <stdio.h>
#define N 20
main()
{
    int t[N], plithos, x, i, j, k;
    plithos=0;
    for (i=0; i<N; i++)
    {
        scanf("%d", &x);
        if (plithos == 0)
        {
            t[plithos]=x;
            plithos++;
        }
        else
        {
            j=0;
            while ((x != t[j]) && (j < plithos - 1))
                j++;
            if (x == t[j])
            {
                for (k=j+1; k < plithos; k++)
                    t[k-1]=t[k];
                plithos--;
            }
            else
            {
                for (k=plithos;k>0;k--)
                    t[k]=t[k-1];
                t[0]=x;
                plithos++;
            }
        }
    }
    for (i=0;i < plithos; i++)
        printf("%d\n",t[i]);
}
```