

Άσκηση 3.6



Να γραφεί μια συνάρτηση με όνομα **overweight** και ορίσματα :

- ένα μονοδιάστατο πίνακα ακεραίων αριθμών n θέσεων
- Έναν ακέραιο αριθμό m

Σε κάθε θέση του πίνακα, αντιστοιχεί ένα βάρος που είναι αντιστρόφως ανάλογο με τον αριθμό της θέσης στον πίνακα. Δηλαδή η 1η θέση έχει βάρος n , η 2η θέση βάρος $n-1$... και η n -οστή θέση βάρος 1 .

Για παράδειγμα αν $n=7$ ο πίνακας θα έχει την παρακάτω δομή:

θέση	1η	2η	3η	4η	5η	6η	7η
βάρος θέσης	7	6	5	4	3	2	1
στοιχείο	5	3	7	6	1	4	9

Η συνάρτηση θα υπολογίζει για κάθε θέση την τιμή: στοιχείο * βάρος θέσης και θα επιστρέφει:

- Τη θέση του πίνακα, έστω k , για την οποία το άθροισμα:
 $\text{στοιχείο}_1 * \text{βάρος θέσης}_1 + \text{στοιχείο}_2 * \text{βάρος θέσης}_2 + \dots + \text{στοιχείο}_k * \text{βάρος θέσης}_k$
γίνεται μεγαλύτερο της τιμής m .
- την τιμή -1 αν δεν βρεθεί τέτοια θέση στον πίνακα

π.χ για το παραπάνω παράδειγμα, και με τιμή $m=100$

το άθροισμα: $5 * 7 + 3 * 6 + 7 * 5 + 6 * 4 = 102$ υπερβαίνει την τιμή $m=100$ στην 4η θέση, άρα η συνάρτηση θα επιστρέψει την τιμή 4 .

Στη συνέχεια να γραφεί πρόγραμμα που να καλεί τη συνάρτηση και να εμφανίζει τα αποτελέσματα.

Άσκηση 3.6 - overweight



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
#define N 7

int overweight(int *A, int m){
    int sum = 0;
    for(int i = 0; i < N; i++){
        sum += A[i] * (N-i);
        if (sum > m)
            return i+1;
    }
    return -1;
}
```

Άσκηση 3.6 - main



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
int main(){
    int A[N] = {5,3,7,6,1,4,9};
    int m = 100, index;
    index = overweight(A, m);
    printf("Το άθροισμα γίνεται μεγαλύτερο του %d, στη θέση: %d \n", m, index);
}
```

Άσκηση 3.7



Μια μονάδα παράγει ισομεγέθεις μεταλλικές ράβδους μήκους d και μετρά τις αποκλίσεις των παραγομένων ράβδων από την επιθυμητή διάσταση d . Θεωρείστε μια σειρά ακεραίων αριθμών, πλήθους N ($N = \text{γνωστό}$) που αντιστοιχεί στις αποκλίσεις που μετρήθηκαν σε ένα δείγμα N ράβδων.

π.χ. για $N = 20$, η σειρά μπορεί να είναι:

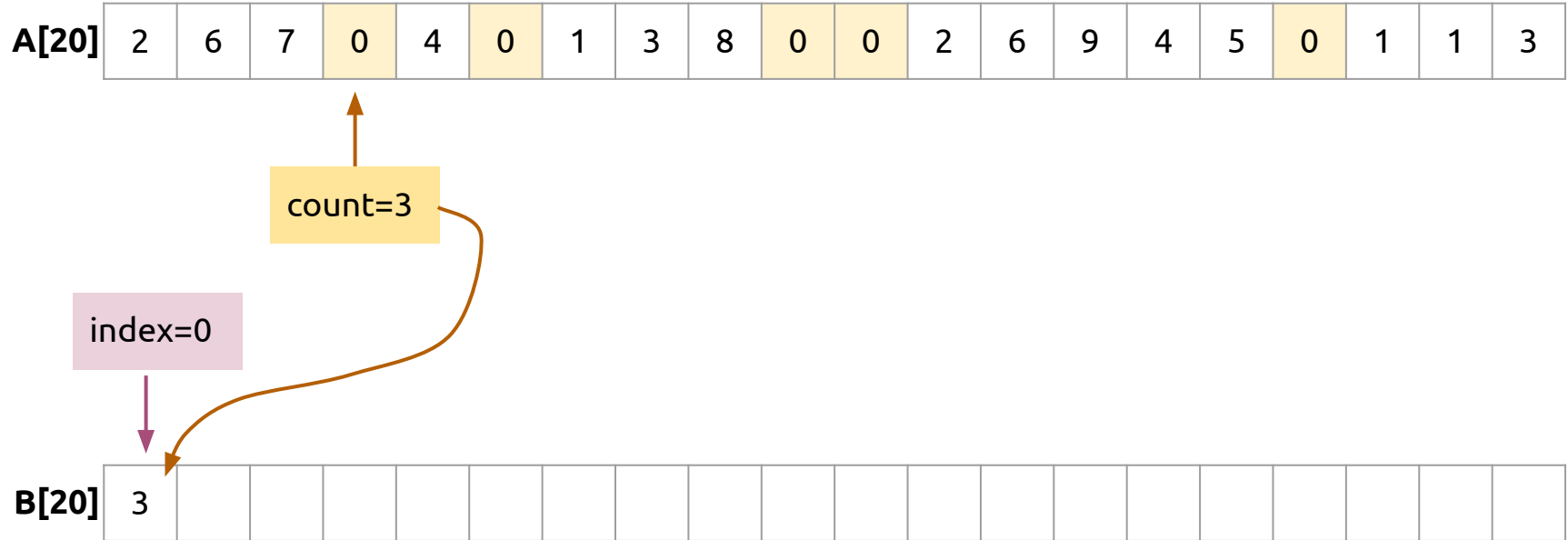
2, 6, 7, 0, 4, 0, 1, 3, 8, 0, 0, 2, 6, 9, 4, 5, 0, 1, 1, 3

Να γράψετε μια συνάρτηση με όνομα `bar_count` που θα δέχεται ως είσοδο ένα μονοδιάστατο πίνακα N θέσεων που αντιστοιχεί στις αποκλίσεις N ράβδων από την επιθυμητή διάσταση και θα επιστρέφει στη `main()`, υπό μορφή πίνακα, το πλήθος των ράβδων που μεσολαβούν ανάμεσα στις ράβδους που έχουν μηδενικές αποκλίσεις.

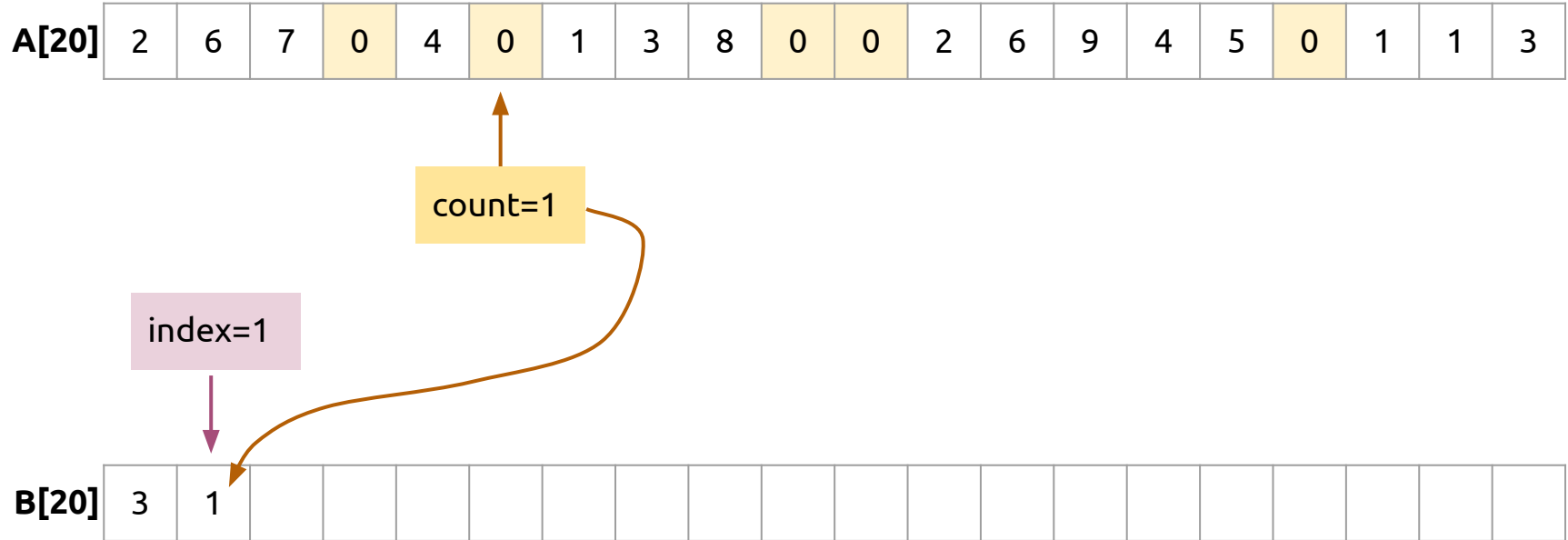
Η συνάρτηση θα καλείται από τη `main()`.

(Δηλαδή για το παράδειγμα θα επιστρέφονται οι τιμές: 3, 1, 3, 0, 5, 3).

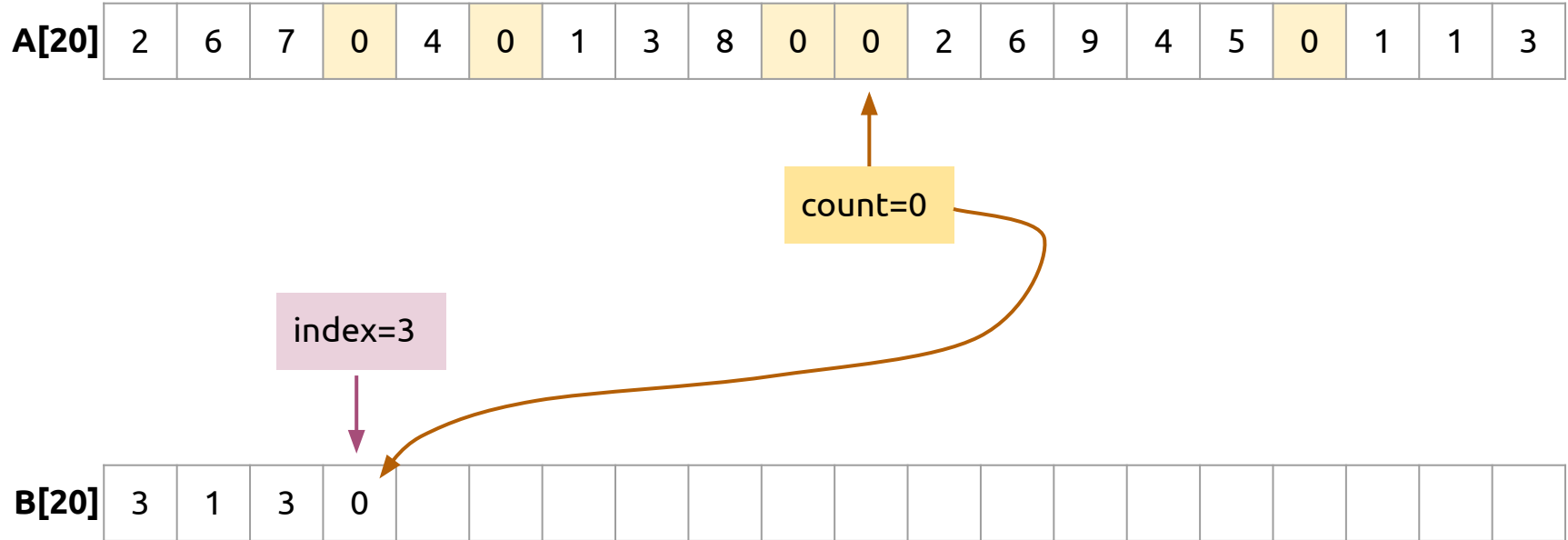
Άσκηση 3.7 - Ανάλυση



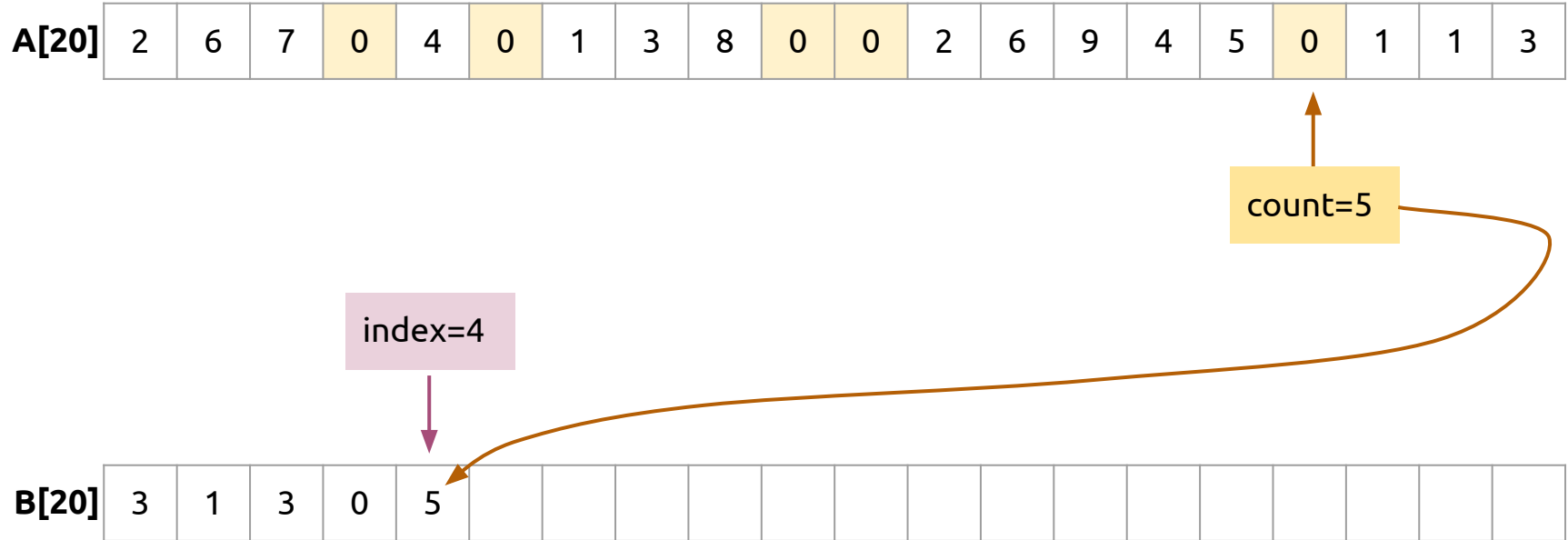
Άσκηση 3.7 - Ανάλυση



Άσκηση 3.7 - Ανάλυση



Άσκηση 3.7 - Ανάλυση



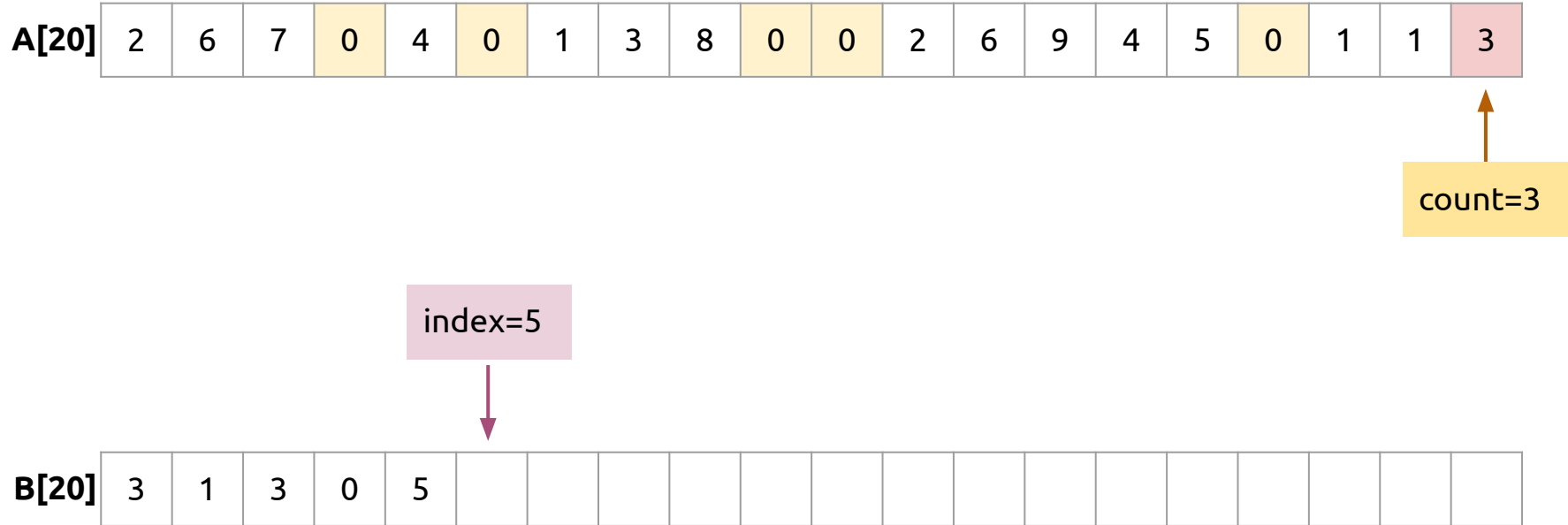
Άσκηση 3.7 - Ανάλυση



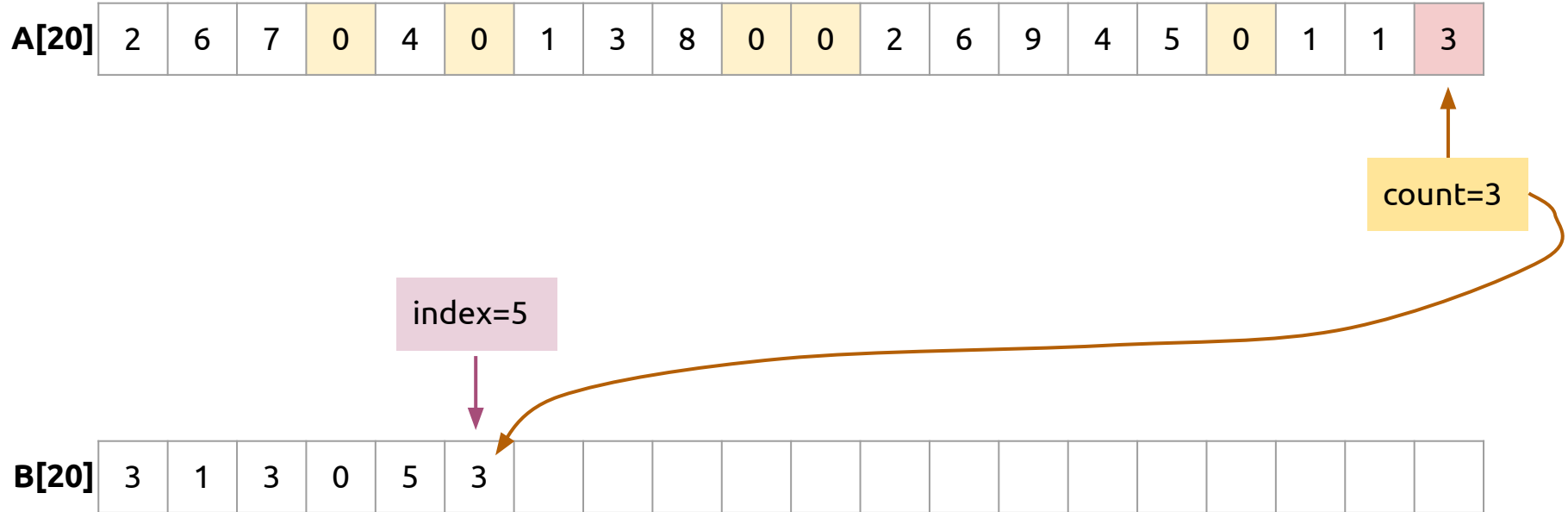
Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ



Άσκηση 3.7 - Ανάλυση



Άσκηση 3.7 - bar_count



```
#include <stdio.h>
#define N 20

int bar_count(int *A, int *B){
    int count, index;

    count = 0;
    index = 0;

    for(int i = 0; i < N; i++){
        if(A[i] != 0){
            count++;
        }
        else{
            B[index] = count;
            count = 0;
            index++;
        }
    }
}
```

```
    if(count != 0){
        B[index] = count;
    }

    return index;
}
```

Άσκηση 3.7 - main



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
int main(){

    int A[N] = {2,6,7,0,4,0,1,3,8,0,0,2,6,9,4,5,0,1,1,3};
    int B[N], index;

    index = bar_count(A, B);

    for(int i = 0; i <= index; i++)
        printf("%d ", B[i]);
}
```