

Εισαγωγή στον Προγραμματισμό

Μονοδιάστατοι Πίνακες

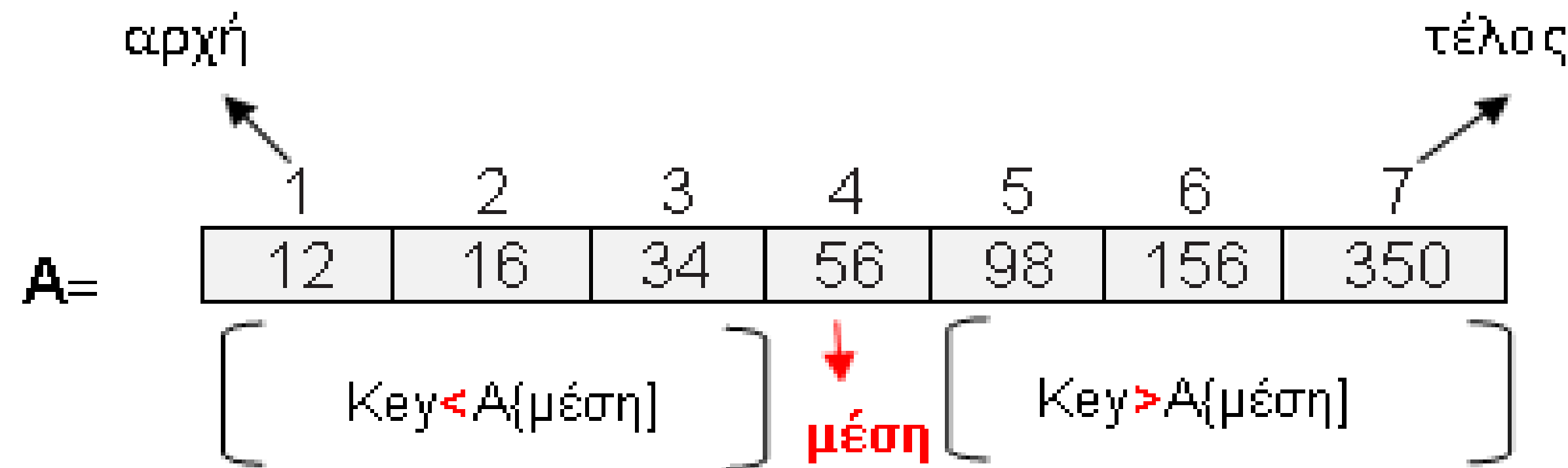
Φροντιστήριο

Αριστείδης Ηλίας

...Μονοδιάστατοι Πίνακες...

Άσκηση 14- Δυαδική αναζήτηση (Μόνο για ταξινομημένους πίνακες)...

- Η δυαδική αναζήτησης είναι αποτελεσματικότερη από την σειριακή γιατί χρειάζεται πολύ λιγότερο χρόνο εκτέλεσης- κάτω από το μισό χρόνο- από ότι η σειριακή αναζήτηση. Απαραίτητη προϋπόθεση είναι ότι ο πίνακας πρέπει να είναι ταξινομημένος ως προς το στοιχείο που αναζητούμε. Η μέθοδος αυτή στηρίζεται στη συνεχή διαίρεση του πίνακα σύμφωνα με τα ακόλουθα βήματα:
 - Ελέγχουμε αν το ζητούμενο στοιχείο είναι ίσο με το στοιχείο του πίνακα που βρίσκεται τη μεσαία θέση.
 - Αν όχι, ελέγχουμε αν το ζητούμενο στοιχείο είναι $<$ με το στοιχείο του πίνακα που βρίσκεται τη μεσαία θέση. Αν ισχύει οπότε αλλάζουμε το τέλος του πίνακα (δηλαδή η αναζήτηση γίνεται μόνο στο 1ο μισό του πίνακα). Διαφορετικά αλλάζουμε την αρχή του πίνακα (δηλαδή ψάχνουμε μόνο στο 2ο μισό του πίνακα).
 - Επαναλαμβάνονται τα προηγούμενα βήματα μέχρι να βρούμε το ζητούμενο στοιχείο. Το ζητούμενο στοιχείο δεν υπάρχει όταν μετά από τις διαδοχικές αυξομειώσεις των άκρων του πίνακα μέχρι να ισχύσει αρχή $>$ τέλος.
- Η παραπάνω διεργασία στηρίζεται ότι ο πίνακας είναι ταξινομημένος κατ' αύξουσα σειρά, π.χ.



- Αντίστοιχα γίνεται και όταν ο πίνακας είναι ταξινομημένος κατά φθίνουσα σειρά.

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

```
int A[10]={21, 32, 43, 54, 65, 76, 79, 88, 99, 110}, key, arxi=0, telos=10, mesi;
```

```
int main() {
```

```
    printf("Δώσε έναν ακέραιο αριθμό:");
```

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

```
    mesi=(arxi+telos)/2;
```

```
    while(arxi<=telos && A[mesi]!=key) {
```

```
        if(key<A[mesi]) telos=mesi-1;
```

```
        else arxi=mesi+1;
```

```
        mesi=(arxi+telos)/2;
```

```
    }
```

```
    if(A[mesi]==key) printf("Το στοιχείο %d βρέθηκε στη θέση %d.\n", key, (mesi+1));
```

```
    else printf("Το στοιχείο δε βρέθηκε στον πίνακα.\n");
```

```
    return 0;
```

```
}
```

...Μονοδιάστατοι Πίνακες...

Άσκηση 15- Ταξινόμηση ευθείας ανταλλαγής (straight exchange sort) ή φυσαλίδας (bubblesort)...

- Γίνονται διαδοχικές προσπελάσεις στον πίνακα και μετακινείται το μικρότερο κλειδί της ακολουθίας προς το αριστερό (ή δεξιό) άκρο του πίνακα.
- Αν ο πίνακας θεωρηθεί σε κατακόρυφη θέση αντί σε οριζόντια και οι ακέραιοι θεωρηθούν ως φυσαλίδες (bubbles) σε μία δεξαμενή νερού με βάρη σύμφωνα με την τιμή τους, τότε κάθε προσπέλαση στον πίνακα έχει ως αποτέλεσμα την άνοδο της φυσαλίδας (**αντιμετάθεση**) στο κατάλληλο επίπεδο βάρους.
- Η μέθοδος είναι γνωστή ως ταξινόμηση φυσαλίδας (bubblesort).

i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9
52	5	5	5	5	5	5	5
12	52	10	10	10	10	10	10
71	12	52	12	12	12	12	12
56	71	12	52	19	19	19	19
5	56	71	19	52	45	45	45
10	10	56	71	45	52	52	52
19	19	19	56	71	56	56	56
90	45	45	45	56	71	71	71
45	90	90	90	90	90	90	90

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

```
int i, j, temp, A[10]={32, 21, 54, 43, 76, 65, 110, 79, 88, 99};
```

```
int main() {  
    for(i=1; i<10; i++) {  
        for(j=9; j>=i; j--) {  
            if(A[j-1]>A[j]) {  
                temp=A[j-1];  
                A[j-1]=A[j];  
                A[j]=temp;  
            }  
        }  
    }  
    for(i=0; i<10; i++) printf("A[%d]= %d\n", i, A[i]);  
    return 0;  
}
```

...Μονοδιάστατοι Πίνακες...

Παράλληλοι πίνακες

- Όταν έχουμε εγγραφές με πεδία διαφορετικού τύπου οι οποίες καταχωρούνται σε πίνακες τότε κάθε πεδίο είναι ένας διαφορετικός πίνακας, εξαιτίας του διαφορετικού τύπου δεδομένων που έχει υποχρεωτικά κάθε πίνακας
- Οι πίνακες αυτοί συνδέονται μεταξύ τους με τέτοιο τρόπο ώστε η κάθε εγγραφή να έχει όλα τα στοιχεία της με την ίδια τιμή δείκτη, δηλαδή να βρίσκονται στην ίδια θέση στους πίνακες
- Π.χ. αν έχουμε εγγραφές που αφορούν στοιχεία μισθοδοσίας υπαλλήλων μιας εταιρείας:

ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΟΙΚΟΓΕΝΙΑΚΗ ΚΑΤΑΣΤΑΣΗ	ΑΡΙΘΜΟΣ ΠΑΙΔΙΩΝ	ΜΙΣΘΟΣ	ΚΡΑΤΗΣΕΙΣ
ΠΑΠΠΑΣ	ΓΕΩΡΓΙΟΣ	ΕΓΓΑΜΟΣ	2	1,400	10%
ΓΕΩΡΓΙΟΥ	ANNA	ΑΓΑΜΗ	0	1,150	20%
ΝΑΝΟΥ	ΣΟΦΙΑ	ΕΓΓΑΜΗ	1	1,520	16%
...	...	...	...	...	...

- Τα παραπάνω δεδομένα μπορούμε να τα τοποθετήσουμε σε 6 διαφορετικούς μονοδιάστατους πίνακες, γιατί άλλα στοιχεία είναι χαρακτήρες και άλλα αριθμοί
- Επίσης, η τοποθέτηση των δεδομένων θα γίνει με τέτοιο τρόπο, ώστε το επώνυμο, όνομα, οικογενειακή κατάσταση, αριθμός παιδιών, μισθός και κρατήσεις ενός υπαλλήλου να βρίσκονται στην ίδια γραμμή σ' όλους τους πίνακες

Μια εταιρεία ταχυμεταφορών χρεώνει τους πελάτες της ανάλογα με τον προορισμό της επιστολής και του βάρους της. Η βασική τιμή εξαρτάται από τον προορισμό ο οποίος καθορίζεται ανάλογα με τον ταχυδρομικό κωδικό που αναγράφεται στην διεύθυνση της επιστολής. Η εταιρεία διατηρεί σε έναν πίνακα με όνομα TK τους 1000 ταχυδρομικούς κωδικούς και σε έναν δεύτερο XR την χρέωση (σε ευρώ) για κάθε έναν από τους 1000 ταχυδρομικούς κωδικούς. Αν μια επιστολή προορίζεται για έναν ταχυδρομικό κωδικό που βρίσκεται εκτός λίστας τότε η βασική τιμή είναι 5 ευρώ. Επίσης ανάλογα με το βάρος της επιστολής η χρέωση ορίζεται σύμφωνα με τον επόμενο πίνακα:

Βάρος επιστολής	Χρέωση
έως και 1000 γρ.	+0% επί της βασικής τιμής
από 1001 έως και 5000 γρ.	+25% επί της βασικής τιμής
από 5001 γρ. και πάνω	+35% επί της βασικής τιμής

Να γίνει πρόγραμμα που:

- Θα δημιουργεί τους δύο πίνακες με έναν τρόπο της επιλογής σας
- Θα διαβάζει τον ταχυδρομικό κωδικό και το βάρος της επιστολής (σε γραμμάρια). Ο ταχυδρομικός κωδικός πρέπει να είναι θετικός πενταψήφιος αριθμός αλλιώς θα πρέπει να ζητείται εκ νέου η καταχώρησή του.
- Να υπολογίζει και εμφανίζει την χρέωση που προκύπτει σύμφωνα με τον προορισμό και το βάρος της επιστολής.

...Μονοδιάστατοι Πίνακες...

...Άσκηση 16- Παράλληλοι πίνακες

```
#include <stdio.h>
int i, TK[1000], taxkod, done=0, proor=-1;
float XR[1000], varos, xreosi;
int main(){
    for(i=0; i<1000; i++){
        TK[i]=10000+i;
        XR[i]=1+(float)i/100;
    }
    do{
        printf("Δώσε ταχυδρομικό κώδικα (5 ψηφία):");
        scanf("%d", &taxkod);
    }while(taxkod<9999 || taxkod>99999);
    i=0;
    while(i<=1000 && done==0){
        if(TK[i]==taxkod){
            proor=i;
            done=1;
        } else i++;
    }
    if(proor==-1){
        printf("Δώσε το βάρος της επιστολής:");
        scanf("%f", &varos);
        if (varos<=1000) xreosi=5;
        else if(varos>1000 && varos<5000) xreosi=5*1.25;
        else xreosi=5*1.35;
    } else {
        xreosi=XR[proor];
    }
    printf("Το κόστος της επιστολής σας είναι %f euro\n", xreosi);
    return 0;
}
```

...Μονοδιάστατοι Πίνακες...

Άσκηση 17- Παράλληλοι πίνακες και πίνακες χαρακτήρων...

Να γραφεί πρόγραμμα το οποίο διαβάζει τα ονόματα 20 αεροπορικών εταιρειών (θεωρήστε ότι αντί για πλήρες όνομα θα χρησιμοποιήσουμε έναν χαρακτήρα, για με πλήρες όνομα χρειαζόμαστε δισδιάστατο πίνακα) και τις αντίστοιχες εισπράξεις τους. Να τυπώνει τα ονόματα των εταιρειών που έχουν εισπράξεις περισσότερες από τον μέσο όρο, αυτές που έχουν εισπράξεις κάτω από το μέσο όρο, να βρίσκει και να τυπώνει την εταιρεία με τις λιγότερες και με τις περισσότερες εισπράξεις.

- Μεθοδολογία

- Το όνομα και η είσπραξη κάθε εταιρείας καταχωρούνται σε διαφορετικούς πίνακες που είναι παράλληλοι. Οπότε τα δεδομένα διαβάζονται ταυτόχρονα.
- Βρίσκουμε το μέσο όρο από τον πίνακα των εισπράξεων.
- Ελέγχουμε κάθε είσπραξη αν είναι μεγαλύτερη του μέσου όρου είσπραξης και τυπώνουμε το αντίστοιχο όνομα εταιρείας.
- Ελέγχουμε κάθε είσπραξη αν είναι μικρότερη του μέσου όρου είσπραξης και τυπώνουμε το αντίστοιχο όνομα εταιρείας.
- Εύρεση μέγιστου και το όνομα εταιρείας που έχει τις μέγιστες.
- Εύρεση ελάχιστου και το όνομα εταιρείας που έχει τις ελάχιστες.

...Μονοδιάστατοι Πίνακες...

...Άσκηση 17- Παράλληλοι πίνακες και πίνακες χαρακτήρων...

```
#include <stdio.h>
#define N 20
int i, check, minbudgthesis, maxbudgthesis;
char compname, NAM[N];
float BUDG[N], sum=0, averag, minbudg, maxbudg;
int main(){
    for(i=0; i<N; i++){
        check=0;
        do{
            printf("Δώσε το όνομα της εταιρίας (μόνο ένας λατινικός χαρακτήρας):");
            scanf(" %c", &compname);
            if((compname>='a' && compname<='z') || (compname>='A' && compname<='Z')) check=1;
        }while(check!=1);
        NAM[i]=compname;
        printf("Δώσε τις εισπράξεις της:");
        scanf("%f", &BUDG[i]);
        sum+=BUDG[i];
    }
    averag=(float)sum/N;
    for(i=0; i<N; i++){
        if(BUDG[i]<averag) printf("Η εταιρία %c έχει εισπράξεις κάτω από το μέσο όρο.\n", NAM[i]);
        if(BUDG[i]>averag) printf("Η εταιρία %c έχει εισπράξεις πάνω από το μέσο όρο.\n", NAM[i]);
    }
    minbudg=BUDG[0];
    minbudgthesis=0;
    maxbudg=BUDG[0];
    maxbudgthesis=0;
    for(i=0; i<N; i++){
        if(minbudg>BUDG[i]){
            minbudg=BUDG[i];
            minbudgthesis=i;
        }
        if(maxbudg<BUDG[i]){
            maxbudg=BUDG[i];
            maxbudgthesis=i;
        }
    }
    printf("Ο μέσος όρος εισπράξεων είναι %f euro.\n", averag);
    printf("Η εταιρία %c εμφανίζει τις λιγότερες εισπράξεις που είναι %f euro.\n", NAM[minbudgthesis], minbudg);
    printf("Η εταιρία %c εμφανίζει τις περισσότερες εισπράξεις που είναι %f euro.\n", NAM[maxbudgthesis], maxbudg);
    return 0;
}
```

...Μονοδιάστατοι Πίνακες...

...Άσκηση 17- Παράλληλοι πίνακες και πίνακες χαρακτήρων

```
#include <stdio.h>
#define N 20
int i, check, minbudgthesis=0, maxbudgthesis=0;
char compname, NAM[N];
float BUDG[N], sum=0, averag, minbudg, maxbudg;
int main(){
    for(i=0; i<N; i++){
        check=0;
        do{
            printf("Δώσε το όνομα της εταιρίας (μόνο ένας λατινικός χαρακτήρας):");
            scanf(" %c", &compname);
            if((compname>='a' && compname<='z') || (compname>='A' && compname<='Z')) check=1;
        }while(check!=1);
        NAM[i]=compname;
        printf("Δώσε τις εισπράξεις της:");
        scanf("%f", &BUDG[i]);
        if(i==0){
            minbudg=BUDG[i];
            maxbudg=BUDG[i];
        }
        sum+=BUDG[i];
        if(minbudg>BUDG[i]){
            minbudg=BUDG[i];
            minbudgthesis=i;
        }
        if(maxbudg<BUDG[i]){
            maxbudg=BUDG[i];
            maxbudgthesis=i;
        }
    }
    averag=(float)sum/N;
    for(i=0; i<N; i++){
        if(BUDG[i]<averag) printf("Η εταιρία %c έχει εισπράξεις κάτω από το μέσο όρο.\n", NAM[i]);
        if(BUDG[i]>averag) printf("Η εταιρία %c έχει εισπράξεις πάνω από το μέσο όρο.\n", NAM[i]);
    }
    printf("Ο μέσος όρος εισπράξεων είναι %f euro.\n", averag);
    printf("Η εταιρία %c εμφανίζει τις λιγότερες εισπράξεις που είναι %f euro.\n", NAM[minbudgthesis], minbudg);
    printf("Η εταιρία %c εμφανίζει τις περισσότερες εισπράξεις που είναι %f euro.\n", NAM[maxbudgthesis], maxbudg);
    return 0;
}
```

Κατασκευάστε πρόγραμμα να υπολογίζει το μέσο όρο στοιχείων δεδομένου πίνακα.

```
#include <stdio.h>

float findAverage(int [], int);

int main(void) {
    int score[5] = {20, 88, 77, 75, 85};
    int papers = 5;
    float avg = findAverage(score, papers);
    printf("Average: %f\n", avg);
    return 0;
}

float findAverage(int arr[], int size) {
    int i;
    float sum = 0, avg = 0;
    for (i = 0; i < size; i++) {
        sum += arr[i];
    }
    avg = sum / size;
    return avg;
}
```

Αναζήτηση του στοιχείου (key) σε ένα πίνακα A[10]. Εάν υπάρχει (ή στην πρώτη εμφάνιση του) να τυπώνεται η θέση που βρέθηκε, αλλιώς να τυπώνεται το μήνυμα "Δε βρέθηκε".

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

```
int findElement(int arr[], int n, int key){  
    int i;  
    for (i = 0; i < n; i++)  
        if (arr[i] == key)  
            return i;  
    return -1;  
}
```

```
int main(){  
    int arr[]={21, 32, 43, 54, 65, 76, 32, 88, 99, 110};  
    int n=sizeof(arr)/sizeof(arr[0]);  
    printf("N=%d", n);  
    int key;  
  
    printf("Δώσε έναν ακέραιο αριθμό:");  
    scanf("%d", &key);  
    int position=findElement(arr, n, key);  
    if (position==-1)  
        printf("Δε βρέθηκε\n");  
    else  
        printf("Το στοιχείο βρέθηκε στη θέση %d\n", position + 1 );  
    return 0;  
}
```

...Μονοδιάστατοι Πίνακες και Χρήση Συναρτήσεων...

Άσκηση 20- Εισαγωγή σε νέα θέση στο τέλος του πίνακα

Κατασκευάστε πρόγραμμα το οποίο να δημιουργεί έναν πίνακα όσων θέσεων επιθυμείτε, να διαβάζει ένα νέο στοιχείο και να το προσθέτει στο τέλος του (αυξάνοντας κατά μια τις θέσεις).

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

```
int insertSorted(int arr[], int n, int key, int capacity){  
    if (n >= capacity) return n;  
    arr[n] = key;  
    return (n + 1);  
}
```

```
int main(){  
    int i, meg, stoixeio;  
  
    printf("Δώσε μέγεθος αρχικού πίνακα:");  
    scanf("%d", &meg);  
    int arr[meg];  
    for(i=0; i<meg; i++){  
        printf("Δώσε στοιχείο αρχικού πίνακα (ακέραιο αριθμό):");  
        scanf("%d", &stoixeio);  
        arr[i]=stoixeio;  
    }  
    int capacity = sizeof(arr) / sizeof(arr[0]);  
    int n = capacity-1;  
    int key;  
    printf("Δώσε τον ακέραιο αριθμό που θα εισαχθεί:");  
    scanf("%d", &key);  
    printf("\n Before Insertion: ");  
    for (i = 0; i < n; i++) printf("%d  ", arr[i]);  
    n = insertSorted(arr, n, key, capacity);  
    printf("\n After Insertion: ");  
    for (i = 0; i < n; i++) printf("%d  ",arr[i]);  
    printf("\n");  
    return 0;
```

...Μονοδιάστατοι Πίνακες και Χρήση Συναρτήσεων...

Άσκηση 21- Διαγραφή στοιχείου και θέσης σε πίνακα

Κατασκευάστε πρόγραμμα το οποίο να διαβάζει έναν αριθμό και να διαγράφει από δεδομένο πίνακα τον αριθμό και τη θέση που βρίσκεται (μειώνοντας κατά μια τις θέσεις).

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

```
int findElement(int arr[], int n, int key);
```

```
int deleteElement(int arr[], int n, int key){
    int pos = findElement(arr, n, key);
    if (pos == - 1){
        printf("Το στοιχείο δε βρέθηκε\n");
        return n;
    }
    int i;
    for (i = pos; i < n - 1; i++) arr[i] = arr[i + 1];
    return n - 1;
}
```

```
int findElement(int arr[], int n, int key){
    int i;
    for (i = 0; i < n; i++) if (arr[i] == key) return i;
    return - 1;
}
```

```
int main(){
    int i;
    int arr[] = {1, 5, 3, 4, 2};
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 3;
    printf("Αρχικός πίνακας:\n");
    for (i = 0; i < n; i++) printf("%d  ", arr[i]);
    n = deleteElement(arr, n, key);
    printf("\nΠίνακας μετά τη διαγραφή\n");
    for (i = 0; i < n; i++) printf("%d  ", arr[i]);
    return 0;
}
```

Προτεινόμενες Ασκήσεις (Μη Βαθμολογούμενες)

1. Μια εταιρεία εξοπλίζει έναν αθλητικό αγώνα με σύστημα χρονομέτρησης. Το σύστημα δημιουργεί δύο παράλληλους πίνακες, τον πίνακα ATHL, με τον αριθμό κάθε αθλητή, και τον πίνακα CLASS, του οποίου η πρώτη θέση περιέχει τον χρόνο που χρειάστηκε να τερματίσει ο πρώτος αθλητής (σε δευτερόλεπτα) και κάθε επόμενη θέση περιέχει τη διαφορά του συγκεκριμένου αθλητή από τον προηγούμενο. Να αναπτυχθεί πρόγραμμα που, με δεδομένους τους παραπάνω πίνακες για έναν αγώνα 10000 μέτρων όπου συμμετείχαν 20 αθλητές, θα:
 - α. διαβάσει τον αριθμό ενός αθλητή και θα εκτυπώνει τον χρόνο που χρειάστηκε για να τερματίσει,
 - β. εκτυπώνει τον χρόνο που διήρκεσε η κούρσα, και
 - γ. εκτυπώνει το πλήθος των αθλητών που είχαν τερματίσει στα μισά της κούρσας.
2. Να γράψετε πρόγραμμα το οποίο να διαβάσει ένα πίνακα 20 πραγματικών αριθμών και να εμφανίζει τους 5 μικρότερους.
3. Μία ακτιβιστική οργάνωση διαθέτει στοιχεία για το ποσοστό δασικών πυρκαγιών για 20 διαφορετικές χώρες (χώρα 1, 2, ..., 20). Αποφάσισε να διοργανώσει εκδήλωση διαμαρτυρίας στις 5 χώρες που έχουν το υψηλότερο ποσοστό. Να κατασκευάσετε πρόγραμμα που θα εμφανίζει τους αριθμούς των 5 αυτών χωρών στις οποίες θα διοργανωθούν οι εκδηλώσεις.
4. Σε μια εκδήλωση έχει καταρτιστεί λίστα καλεσμένων. Σε πίνακα INV καταχωρούνται οι αριθμοί της πρόσκλησης των καλεσμένων και σε έναν πίνακα TABL καταχωρείται ο αριθμός του τραπέζιού που θα κάτσει κάθε καλεσμένος. Τα τραπέζια διαθέτουν 10 θέσεις και το συνολικό πλήθος των καλεσμένων είναι 1500. Να αναπτυχθεί πρόγραμμα θα:
 - i. δημιουργεί του παραπάνω πίνακες, έτσι ώστε οι πίνακες INV και TABL να περιέχουν τα στοιχεία τους μη συνεχόμενα και μη ταξινομημένα(θα πρέπει να σχεδιάσετε έναν αλγόριθμο και δεν αναφέρουμε τυχαίο τρόπο για να μη χρειαστεί να χρησιμοποιήσετε αντίστοιχη συνάρτηση παραγωγής τυχαίων αριθμών),
 - ii. διαβάσει τον αριθμό μιας πρόσκλησης ενός καλεσμένου και θα εκτυπώνει το τραπέζι στο οποίο έχει τοποθετηθεί,
 - iii. διαβάσει το αριθμό ενός τραπέζιού και να εκτυπώνει τους αριθμούς των προσκλήσεων της λίστας των ατόμων που κάθονται σε αυτό,
 - iv. εκτυπώνει ταξινομημένα σε αύξουσα σειρά το όνομα κάθε καλεσμένου και το τραπέζι του,
 - v. θα εκτυπώνει τους αριθμούς των προσκλήσεων των καλεσμένων της εκδήλωσης ανά τραπέζι.
5. Να κατασκευάσετε πρόγραμμα που να καταχωρεί, για μια πόλη, τις θερμοκρασίες από 28 συνεχόμενες ημέρες (4 εβδομάδων) την ίδια ώρα και στο ίδιο σημείο της πόλης. Να υπολογίσετε ποια ημέρα (π.χ. Δευτέρα) ήταν πιο θερμή κατά μέσο όρο σε αυτό το διάστημα.

Ευχαριστώ

Ερωτήσεις?