



Πανεπιστήμιο Θεσσαλίας
Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική

Εισαγωγή στον Προγραμματισμό
Ακαδημαϊκό Έτος 2017-2018
<http://emarkou.users.uth.gr/greek/teach/progintro>
Ε. Μάρκου

10η Σειρά Ασκήσεων

Άσκηση 11.

Δίνονται οι εξής συναρτήσεις για έναν πίνακα θετικών ακεραίων:

```
int initialize (int C[]);  
// Πριν:  
// Μετά: δημιουργεί τον πίνακα C και επιστρέφει 1  
{  
    return assign(C,0,GUARD);  
}  
  
int assign(int C[], int pos, int x);  
// Πριν:  
// Μετά: αν η θέση pos είναι νόμιμη, τοποθετεί το στοιχείο x στην θέση pos του  
// πίνακα C και επιστρέφει 1, αλλιώς επιστρέφει 0  
{  
    if ((pos>=0) && (pos <= MAX_LENGTH))  
    {  
        C[pos]=x;  
        return 1;  
    }  
    else  
        return 0;  
}  
  
int extract(int C[], int pos, int *x);  
// Πριν: δημιουργία του πίνακα C  
// Μετά: αν η θέση pos είναι νόμιμη, βάζει την τιμή του στοιχείου C[pos] στο x  
// και επιστρέφει 1, αλλιώς επιστρέφει 0  
{  
    if ((pos>=0) && (pos <= MAX_LENGTH))  
    {  
        *x = C[pos];  
        return 1;  
    }  
    else  
        return 0;  
}
```

Να υλοποιήσετε τις παρακάτω συναρτήσεις:

```
int length(int C[]);
// Πριν: δημιουργία του πίνακα C
// Μετά: επιστρέφει το μήκος του πίνακα C

int insert(int C[], int x);
// Πριν: δημιουργία του πίνακα C
// Μετά: εάν length(C) < MAX_LENGTH εισάγει το στοιχείο x στο τέλος του
// πίνακα C και επιστρέφει 1, αλλιώς επιστρέφει 0.

int search(int C[], int x, int *pos);
// Πριν: δημιουργία του πίνακα C
// Μετά: αν το στοιχείο x υπάρχει στον πίνακα C, βάζει στο pos τη θέση της
// πρώτης του εμφάνισης και επιστρέφει 1, αλλιώς επιστρέφει 0

int delete(int C[], int x);
// Πριν: δημιουργία του πίνακα C
// Μετά: αν υπάρχει το στοιχείο x στον πίνακα C το διαγράφει και επιστρέφει 1,
// αλλιώς επιστρέφει 0

int min(int C[], int *x, int *pos);
// Πριν: δημιουργία του πίνακα C
// Μετά: αν ο πίνακας C έχει τουλάχιστον 1 στοιχείο, βάζει στο x το ελάχιστο
// στοιχείο και στο pos τη θέση της πρώτης του εμφάνισης στον C και
// επιστρέφει 1, αλλιώς επιστρέφει 0

int max(int C[], int *x, int *pos);
// Πριν: δημιουργία του πίνακα C
// Μετά: αν ο πίνακας C έχει τουλάχιστον 1 στοιχείο, βάζει στο x το μέγιστο
// στοιχείο και στο pos τη θέση της πρώτης του εμφάνισης στον C και
// επιστρέφει 1, αλλιώς επιστρέφει 0
```

Να φτιάξετε ένα πρόγραμμα που θα παρουσιάζει στην οθόνη επαναληπτικά ένα μενού στο οποίο ο χρήστης θα μπορεί να διαλέγει τις λειτουργίες:

- 1) Insert.
- 2) Search.
- 3) Delete.
- 4) Length.
- 5) Min.
- 6) Max.
- 7) Exit.

Στο πρόγραμμά σας θα πρέπει να χρησιμοποιήσετε αποκλειστικά τις παραπάνω συναρτήσεις και θα ξεκινά ως εξής:

```

#include<stdio.h>

#define MAX_LENGTH 100 // μέγιστο πλήθος στοιχείων
#define GUARD -1 // δηλώνει το τέλος του πίνακα

main()
{
    int B[MAX_LENGTH+1];

    ...

```

Παρατηρήσεις:

Μετά από κάθε λειτουργία ο πίνακας θα πρέπει να παραμένει συνεπής με τον ορισμό του. Δηλαδή αν εκτελεστεί η λειτουργία *insert*, στο τέλος του πίνακα θα πρέπει πάντα να υπάρχει η τιμή **-1** και το μήκος του να έχει αυξηθεί κατά **1**. Επίσης, αν εκτελεστεί η λειτουργία *delete* του στοιχείου *x* (το οποίο έστω ότι υπάρχει π.χ., στις θέσεις **5** και **8** του πίνακα), τα στοιχεία του πίνακα θα πρέπει να ολισθήσουν αριστερά έτσι ώστε το νέο μήκος του πίνακα να είναι μειωμένο κατά **2**.

Τα σχόλια ‘*Πριν*’ και ‘*Μετά*’ στις συναρτήσεις παραπάνω, δείχνουν τί προϋποθέτει η συνάρτηση για να λειτουργήσει σωστά, και ποιό είναι το αποτέλεσμα μετά την κλήση της αντίστοιχα. Για την υλοποίηση των συναρτήσεων, θα πρέπει να χρησιμοποιήσετε αποκλειστικά και μόνο τις παραπάνω συναρτήσεις. Απαγορεύεται να κάνετε απ’ ευθείας ανάθεση ή καταχώρηση τιμής σε στοιχείο του πίνακα. Για παράδειγμα η συνάρτηση *length* μπορεί να υλοποιηθεί ως εξής:

```

int length(int C[])
{
    int y, dummy;
    int i=0;

    do
    {
        dummy = extract(C, i, &y);
        i++;
    } while (y != GUARD)

    return (i-1);
}

```

Να επιδειχθεί στον υπεύθυνο του εργαστηρίου την εβδομάδα 8/1/2018 – 12/1/2018