

Θέματα παλαιότερων εξεταστικών για το μάθημα Γλώσσα Προγραμματισμού - Θεωρία

Ποιες είναι οι βασικές δομές που χρησιμοποιεί ο δομημένος προγραμματισμός; Τι γνωρίζετε για τη δομή ακολουθίας;

Η δομή της ακολουθίας, η δομή ελέγχου (ή επιλογής) και η δομή επανάληψης. Σύμφωνα με τη δομή ακολουθίας οι εντολές εκτελούνται ακολουθιακά, δηλαδή μία – μία με τη σειρά που τις γράψαμε, οπότε η σειρά με την οποία τις γράψαμε είναι άκρως σημαντική.

Ποιες είναι οι δομές ελέγχου/επιλογής που χρησιμοποιούμε στην γλώσσα προγραμματισμού C;

Είναι η απλή, η σύνθετη και η πολλαπλή δομή επιλογής. Σε κάποιες περιπτώσεις όταν και μόνο όταν συγκρίνεται στη συνθήκη ελέγχου χαρακτήρας ή ακέραιος μπορώ να χρησιμοποιήσω αντί της πολλαπλής δομής την δομή switch – case

Ποιες είναι οι τρεις (3) δομές επανάληψης που χρησιμοποιούμε στην γλώσσα C;

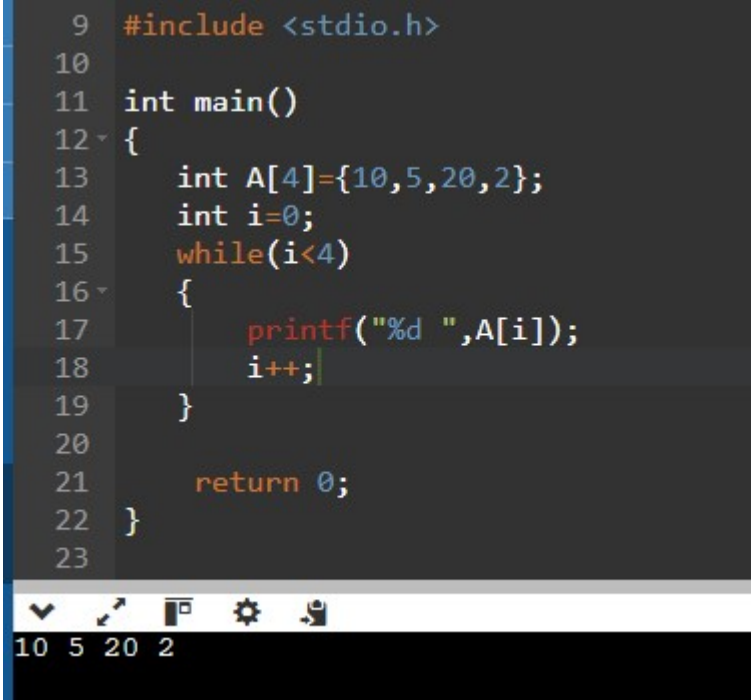
Είναι η for, η while() και η do {} while.

Οι τρεις δομές μπορούν να χρησιμοποιηθούν τόσο για γνωστό όσο και για άγνωστο αριθμό επαναλήψεων.

Να δηλώσετε και να αρχικοποιήσετε έναν πίνακα με όνομα *A* που να περιέχει 4 ακέραιους της επιλογής σας.. Ποια δομή από αυτές που γράψατε στο ερώτημα 1 θα χρησιμοποιήσω για να εμφανίσω τα στοιχεία του πίνακα;

Θα χρησιμοποιήσω μια δομή επανάληψης. Εδώ γίνεται με την `while()`

```
9  #include <stdio.h>
10
11  int main()
12  {
13      int A[4]={10,5,20,2};
14      int i=0;
15      while(i<4)
16      {
17          printf("%d ",A[i]);
18          i++;
19      }
20
21      return 0;
22  }
23
```

The image shows a code editor window with a dark background. The code is written in C and uses a while loop to iterate through an array A containing the values 10, 5, 20, and 2. The loop prints each element followed by a space. Below the code editor, there is a terminal window showing the output of the program: "10 5 20 2". The terminal window has a standard Linux-style prompt and a few icons on the left side.

10 5 20 2

Τι γνωρίζετε για τις μεταβλητές τύπου δείκτη; Τι περιέχουν ως τιμή αυτές οι μεταβλητές; Μπορείτε να δείξετε ένα παράδειγμα; Μπορείτε να κάνετε έναν δείκτη να δείχνει σε μια μεταβλητή;

Οι μεταβλητές τύπου δείκτη είναι μεταβλητές όπως είναι και οι μεταβλητές ακεραίων, δεκαδικών και χαρακτήρων που μάθαμε ως τώρα με τη μόνη διαφορά ότι ως τιμή περιέχουν έναν δεκαεξαδικό αριθμό που είναι η διεύθυνση μιας άλλης μεταβλητής στην μνήμη RAM.

```
9  #include <stdio.h>
10
11  int main()
12  {
13      int a=100;
14      int *pa=&a;
15      printf("%p\n",pa);
16      printf("%d",*pa);
17
18      return 0;
19  }
20
```

0x7ffec4e8584c
100

Δήλωση κι αρχικοποίησης μιας ακεραίας μεταβλητής με όνομα a και αρχική τιμή το 100

Κάνω τον δείκτη pa να δείχνει στην μεταβλητή a. Αυτό γίνεται γιατί παίρνω την διεύθυνση στη RAM της a και την βάζω ως τιμή-περιεχόμενο στην pa

Αυτή είναι η τιμή του δείκτη. Ο δείκτης περιέχει έναν δεκαεξαδικό αριθμό. Είναι η διεύθυνση της μεταβλητής a

Εμφανίζω την τιμή της μεταβλητής a έμμεσα με χρήση του δείκτη pa. Κάνω αποαναφοροποίηση στον δείκτη pa όπως λέγεται.

Πώς λέγεται ο τύπος δεδομένων που μπορεί να ορίσει ο χρήστης στη C και μπορεί να αναπαραστήσει σύνθετες μεταβλητές που υπάρχουν στον πραγματικό κόσμο; Μπορείτε να ορίσετε και εσείς έναν τέτοιο τύπο με πληροφορίες της επιλογής σας;

Η C μας δίνει την δυνατότητα να ορίσουμε μεταβλητές με δικό μας τύπο που να ανταποκρίνονται στις ανάγκες του πραγματικού κόσμου. Αυτό γίνεται με τις structs . Παράδειγμα με δημιουργία μίας δομής struct που κρατάει πληροφορίες για κάποιο άτομο-πρόσωπο . Οι πληροφορίες είναι το όνομα του, το επώνυμο, η πόλη στην οποία μένει και η ηλικία του. Θα δώσω το όνομα Person στην struct γιατί αυτό θεωρώ πιο παραστατικό.

```
struct Person
{
    char firstName[50];
    char lastName[50];
    char city[40];
    int age;
} ;
```

Ποιον αλγόριθμο ταξινόμησης γνωρίζετε; Τι γνωρίζετε για την σειριακή αναζήτηση στοιχείου σε πίνακα και τι γνωρίζετε για την διχοτομική αναζήτηση στοιχείου σε πίνακα; Ποια από τις δύο αναζητήσεις (σειριακή και διχοτομική) εφαρμόζεται υποχρεωτικά σε ταξινομημένο πίνακα;

Υπάρχουν αρκετοί αλγόριθμοι ταξινόμησης:

1. Φυσαλίδας
 2. Εισαγωγής
 3. Επιλογής
 4. Quick sort
 5. Merge sort
- και άλλες....

Ο αλγόριθμος της φυσαλίδας είναι ο πιο εύκολος για εκμάθηση κι ας είναι ο χειρότερος από πλευράς απόδοσης σε σύγκριση με τους άλλους.

Η σειριακή αναζήτηση αναζητά ένα στοιχείο x σειριακά σε έναν πίνακα A . Ξεκινάει από την αρχή του πίνακα A (δηλαδή τη θέση $A[0]$) και πάει θέση-θέση ως το τέλος του πίνακα, συγκρίνοντας κάθε φορά αν το x είναι ίσο με το $A[i]$ στοιχείο.

Η σειριακή αναζήτηση μπορεί να εφαρμοστεί σε ταξινομημένο ή αταξινομητο πίνακα. Ειδικότερα αν ο πίνακας είναι ταξινομημένος ο χρήστης έχει και καλύτερη επιλογή να χρησιμοποιήσει την δυαδική-διχοτομική αναζήτηση που είναι πολύ πιο γρήγορη, ωστόσο η σειριακή αναζήτηση μπορεί να εφαρμοστεί κανονικά, στον ταξινομημένο πίνακα, αν ο χρήστης επιθυμεί να χρησιμοποιήσει μόνο αυτή.

Η διχοτομική αναζήτηση εφαρμόζεται ΜΟΝΟ σε ταξινομημένο πίνακα.

Δίνονται παρακάτω δύο δομές επανάληψης για την εμφάνιση των κελιών ενός πίνακα. Είναι ίδιες ή όχι;

```
int main()
{
    int A[7]={3,1,6,8,2,90,33},i;
    for(i=0;i<7;i++)
    {
        printf("%d ",A[i]);
    }

    i=0;
    while(i<=6)
    {
        printf("%d ",A[i]);
        i++;
    }
}
```

Οι εντολές είναι ακριβώς οι ίδιες. Απλά η δομή `for` έχει όλες τις εντολές σε μία γραμμή ενώ η δομή `while` τις κάνει σε τρεις διαφορετικές γραμμές. Επίσης η συνθήκη ελέγχου στην `for(..i<7)` είναι ίδια με την συνθήκη ελέγχου `i<=6` της `while`