

Δομημένος Προγραμματισμός

Τμήμα Επιχειρηματικού Σχεδιασμού και
Πληροφοριακών Συστημάτων

www.bpis.teicrete.gr

Κανόνες χρήσης προτάσεων επανάληψης

- Σωστή χρήση στηλοθεσίας για εύκολη αναγνωσιμότητα κώδικα
- Χρησιμοποιείται { } για να περικλείετε περισσότερες της μίας πρότασης
- **Αποφεύγετε τη χρήση της πρότασης διακλάδωσης goto**
- Προτιμήστε το βρόχο επανάληψης με συνθήκη εισόδου (while) από τον αντίστοιχο με συνθήκη εξόδου (dowhile)

Κανόνες χρήσης προτάσεων επανάληψης

- Αποφεύγετε κατά το δυνατόν τη χρήση των **break** και **continue** σε βρόχους επανάληψης
- Εκτελέστε λεπτομερή έλεγχο και βεβαιωθείτε ότι κάθε συνθήκη βρόχου επανάληψης οδηγεί στην έξοδο μετά από πεπερασμένες επαναλήψεις, έτσι ώστε να μη δημιουργούνται ατέρμονοι βρόχοι

Ένθετοι βρόχοι (nested loops)

- Ονομάζεται ο βρόχος που περικλείεται σε έναν άλλο
- Ο εσωτερικός βρόχος λογίζεται ως μια πρόταση του εξωτερικού – δηλαδή πρώτα θα εκτελείται ο εσωτερικός βρόχος και μετά η επόμενη επανάληψη του εξωτερικού
- Η C επιτρέπει πολλαπλή ένθεση
- Ο συνολικός αριθμός επαναλήψεων σε έναν πολλαπλό βρόχο είναι το γινόμενο του αριθμού των επαναλήψεων όλων των επιμέρους βρόχων

Ένθετοι βρόχοι (nested loops)

```
for( i=0; i<4; i++)  
{  
    for( j=0; j<3; j++)  
    {  
        printf( “ (%d.%d) ”, i , j );  
    }  
}
```

Μονοδιάστατοι Πίνακες (tables)

Μια συλλογή μεταβλητών ίδιου τύπου οι οποίες είναι αποθηκευμένες σε διαδοχικές θέσεις μνήμης
Τύπους Πινάκων τόσους όσους και οι τύποι δεδομένων
Κύκλος ζωής όπως και οι μεταβλητές
Δήλωση κανονικά πριν την χρήση
Δήλωση πίνακα

```
Τύπος_δεδομένων όνομα_πίνακα[μέγεθος] ;
```

Για την δήλωση έχουμε τρία στοιχεία

τύπος

όνομα

μέγεθος

π.χ. float temp[31];

Αναφορά σε στοιχεία πίνακα

Η αναφορά σε στοιχεία του πίνακα γίνεται σε συνδυασμό με το **όνομα του πίνακα** και με την χρήση **ενός δείκτη** ο οποίος εκφράζει τη σειρά του στοιχείου μέσα στον πίνακα

```
float temp[31];
```

temp[0] → αναφερόμαστε στο πρώτο στοιχείο του πίνακα

temp[1] → αναφερόμαστε στο δεύτερο στοιχείο του πίνακα

temp[30] → αναφερόμαστε στο τελευταίο στοιχείο του πίνακα

Αρχικές τιμές σε πίνακα

Μπορούμε να αναθέσουμε αρχικές τιμές στον πίνακα όπως και στις μεταβλητές

Που τις αναθέτουμε ;

Ακριβώς στην δήλωση όπως με τις μεταβλητές

Για αρχικές τιμές σε πίνακα χρειαζόμαστε τους τελεστές

{ } και **,**

πχ. `float temp[5] = { 1, 2, 4.2, 6, 8 } ;`

πχ. `float temp[5] = { 1, 2, 4.2 };`

Ανάγνωση τιμών στοιχείου πίνακα

Για τις μεταβλητές

```
int main()
{
    int var1;
    printf( "Give a value : " );
    scanf( "%d", &var1 );
    printf( "You gave :%d", var1 );
    return 0;
}
```

Για ένα στοιχείο πίνακα

```
int main()
{
    int var[20];
    printf( "Give a value : " );
    scanf( "%d", &var[0] );
    printf( "You gave :%d", var[0] );
    return 0;
}
```

Ανάγνωση τιμών στοιχείου πίνακα

Για όλα τα στοιχεία του πίνακα

```
int main()  
{  
    int i, var[20];  
    for( i=0; i<20; i++ )  
    {  
        printf( "Give a value :" );  
        scanf( "%d", &var[i] );  
    }  
  
    for( i=0; i<20; i++)  
    {  
        printf( "The %d element is :%d", i, var[i] );  
    }  
    return 0;  
}
```

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

table

Άσκηση

Δημιουργήστε ένα πίνακα 5 θέσεων και αρχικοποιήστε τον με τις παρακάτω τιμές:

10 20 30 40 50

στη συνέχεια τυπώστε τα περιεχόμενα του στην οθόνη

Άσκηση

```
int i, table[5] = {10, 20, 30, 40, 50};  
  
for ( i = 0; i<5; i++)  
    printf("value of table[%d] is %d\n", i, table[i]);
```

Ιδιαιτερότητες σχετικές με Πίνακες

Όταν αποδίδονται αρχικές τιμές μπορεί να παραληφθεί το μέγεθος του πίνακα

```
char pin[] = {'a','b','c','d'};
```

Οι δείκτες των στοιχείων ενός πίνακα ξεκινούν από το 0 και όχι από το 1

Ο δείκτης έχει διαφορετική σημασία στην δήλωση και διαφορετική στην αναφορά.

Στην δήλωση καθορίζει μέγεθος

στην αναφορά προσδιορίζει το στοιχείο αναφοράς

Ορισμός μεγέθους πίνακα

Από το προηγούμενο παράδειγμα αφαιρέστε το μέγεθος του πίνακα στον ορισμό του.

Τι παρατηρείτε;

Χρησιμοποιήστε την `define` για να ορίσετε καθολικά το μέγεθος του πίνακα

Τροποποιήστε το πρόγραμμα σας ώστε να εισάγονται οι τιμές 10-50 αυτόματα

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

table2

Άσκηση

Δημιουργήστε ένα πρόγραμμα όπου ορίζονται
2 πίνακες ακεραίων 5 θέσεων

και αρχικοποιήστε τους με τις παρακάτω τιμές:

Πίνακας 1 → 10 20 30 40 50

Πίνακας 2 → 50 40 30 20 10

Στη συνέχεια αποθηκεύστε σε δύο άλλους πίνακες
diff και **sum** τη διαφορά και το άθροισμα τους.

Χρησιμοποιήστε το αρχείο `table2_sample.c`

```
int i, table1[SIZE] = {10, 20, 30, 40, 50};
int table2[SIZE] = {50, 40, 30, 20, 10};

int sum[SIZE],diff[SIZE];

...

for ( i = 0; i < SIZE; i++)
    sum[i] = table1[i] + table2[i];

for ( i = 0; i < SIZE; i++)
    diff[i] = table1[i] - table2[i];

printf("\nAddition (Table1+Table2): \n");
for ( i = 0; i < SIZE; i++)
    printf("sum[%d] = %d\n", i, sum[i]);

printf("\nDifference (Table1-Table2): \n");
for ( i = 0; i < SIZE; i++)
    printf("diff[%d] = %d\n", i, diff[i]);
```

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

table3

Άσκηση – Σειριακή Αναζήτηση

Γράψτε ένα πρόγραμμα το οποίο

Θα αρχικοποιεί ένα πίνακα **10 θέσεων** τύπου **int** με τις παρακάτω τιμές

10 20 30 40 50 60 70 80 90 100

Θα ζητάει ένα ακέραιο αριθμό από τον χρήστη

Και στη συνέχεια θα τον αναζητάει στους αριθμούς του πίνακα.

- Αν βρεθεί ο αριθμός θα εμφανίζεται η θέση του πίνακα στην οποία βρίσκεται (θέση από 0 έως 9).
- Διαφορετικά θα εμφανίζεται το μήνυμα «Ο arithmos den breθηke»

Βρόχος while

- Βρόχος με συνθήκη εισόδου
- Μπορεί να οδηγείται τόσο από μετρητή όσο και από γεγονός
- Για όσο η συνθήκη είναι αληθής επανέλαβε
- Έλεγχος συνθήκης στην αρχή
- Ο βρόχος while για να έχει λογική υπόσταση θα πρέπει
 - πριν την πρώτη επανάληψη η συνθήκη να είναι αληθής
 - στο μπλόκ εντολών επανάληψης να υπάρχει η δυνατότητα αλλαγής της συνθήκης σε ψευδής
- Επιλέγεται σε περιπτώσεις που δεν είναι γνωστός εκ των προτέρων ο αριθμός των επαναλήψεων

Βρόχος while στην C Δήλωση & Λειτουργία

```
int count = 30;
while( count < 40 )
{
    count++;
    printf( "count is %d\n", count );
}
```

Βρόχος do-while στην C Δήλωση & Λειτουργία

```
int count = 30;  
do  
{  
    count++;  
    printf( "count is %d\n", count );  
}  
while( count < 40 );
```

Νέο Πρόγραμμα

Δημιουργήστε ένα νέο project με τίτλο

table4

Άσκηση

Γράψτε ένα πρόγραμμα το οποίο

Θα δέχεται από το πληκτρολόγιο **συνεχώς** ακέραιους αριθμούς μέχρι να εισαχθούν **5 διαφορετικοί αριθμοί**

Οι αριθμοί θα αποθηκεύονται σε ένα πίνακα

Καθώς διαβάζεται ένας αριθμός, **θα τυπώνεται στην οθόνη μόνο εάν δεν έχει εισαχθεί πριν.**

Μετά την εισαγωγή των 5 διαφορετικών αριθμών θα τυπώνονται στην οθόνη και οι 5 αριθμοί

Πολυδιάστατοι Πίνακες

Πίνακας μιας διάστασης των οποίων τα στοιχεία είναι επίσης πίνακας

```
πχ int array[4][12];
```

Δήλωση πολυδιάστατου πίνακα

Τύπος_Δεδομένων όνομα_πίνακα[πλήθος γραμμών][πλήθος στηλών];

Ο πολυδιάστατος πίνακας αποθηκεύεται στη μνήμη ως μία ακολουθία στοιχείων μίας διάστασης

Για την δήλωση έχουμε τρία στοιχεία

τύπος

όνομα

μέγεθος σε πλήθος γραμμών και στηλών

```
π.χ. int array[10][8];
```

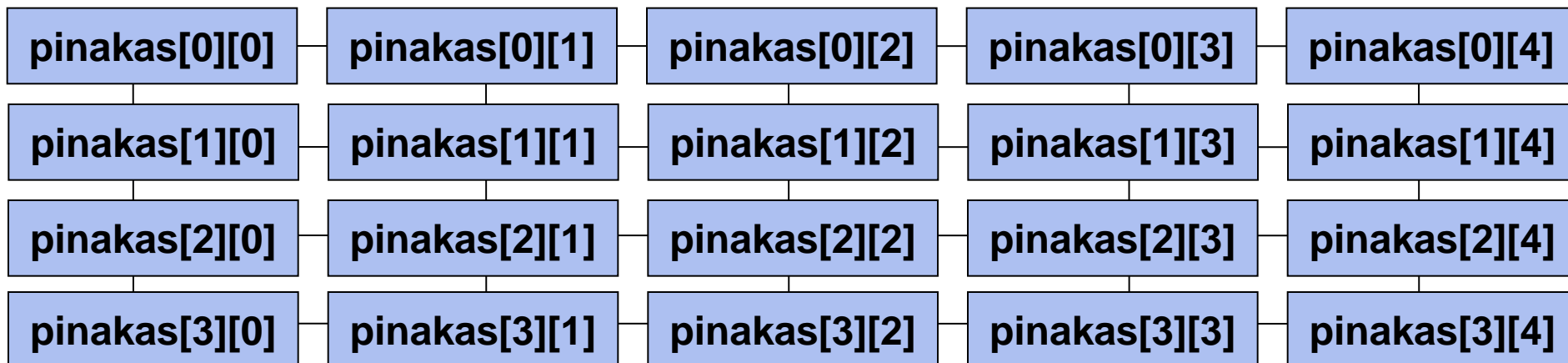
Πολυδιάστατοι Πίνακες

Ο πίνακας 2 διαστάσεων χαρακτηρίζεται από
τον τύπο (int, float, double, char...)

το όνομα

το μέγεθος κάθε διάστασης (γραμμές x στήλες)

Απεικόνιση πίνακα 4 γραμμών και 5 στηλών
π.χ. float pinakas[4][5];



Αρχικές τιμές σε πολυδιάστατο πίνακα

Για αρχικές τιμές σε πίνακα χρειαζόμαστε τους τελεστές `{ }` και `,`

Τα στοιχεία της κάθε γραμμής εσωκλείονται σε `{ }` διαχωρίζοντας τα με `,` και όλες οι γραμμές μαζί μετά σε `{ }`

```
πχ. int temp[5][5] = {      {17,24,1,8,15},
                          {23,5,7,14,16},
                          {4,6,13,20,22},
                          {10,12,19,21,3},
                          {11,18,25,2,9}
                        };
```

Αρχικές τιμές σε πολυδιάστατο πίνακα

Προσοχή

```
temp[5][3] = {      {1,2,3},  
              {4},  
              {5,6,7} };
```

Επίσης

```
int temp[5][3]={1,2,3,4,5,6,7};
```

Παράλειψη της πρώτης διάστασης

```
int temp[][3][2]={   { {1,1},  
                      {0,0},  
                      {1,1} },  
                    { {0,0},  
                      {1,2},  
                      {0,1} },  
                    };
```

→ πίνακας 2x3x2