

Δομημένος Προγραμματισμός

Τμήμα Επιχειρηματικού Σχεδιασμού και
Πληροφοριακών Συστημάτων

www.bpis.teicrete.gr

Παρατηρήσεις

- Η `scanf` είναι για μορφοποιημένη είσοδο δεδομένων και η `printf` είναι για μορφοποιημένη έξοδο.
- Όταν θέλουμε απλά να εισάγουμε κάποιους χαρακτήρες χρησιμοποιούμε τις **`getch`**, **`getche`**, **`getchar`**.
- Όταν θέλουμε απλά να τυπώσουμε κάποιους χαρακτήρες χρησιμοποιούμε την **`putchar`**.

Συναρτήσεις `getche`, `getch`

- Περιγράφονται στην βιβλιοθήκη `conio.h`
- Η `getche` διαβάζει έναν χαρακτήρα από την κύρια είσοδο (πληκτρολόγιο) και στην συνέχεια επιστρέφει την τιμή του, εμφανίζοντας στην οθόνη το πλήκτρο που πατήθηκε
- Η `getch` διαφέρει στο ότι **δεν** εμφανίζει τον πληκτρολογηθέντα χαρακτήρα στην οθόνη

Παράδειγμα με την getch

- Ένα πρόγραμμα που διαβάζει ένα χαρακτήρα από το πληκτρολόγιο και τον εμφανίζει στην οθόνη.

```
#include<stdio.h>
#include<conio.h>
void main(void)
{
    int x;
    x = getch();
    printf( " \n Ο xaraktiras einai %c " ,x);
}
```

Νέο Πρόγραμμα

- Δημιουργήστε ένα νέο project με τίτλο

Getche

Άσκηση

- Πληκτρολογήστε το προηγούμενο παράδειγμα και εκτελέστε το.
- Δοκιμάστε στο προηγούμενο παράδειγμα να χρησιμοποιήσετε την `getch`. Τι παρατηρείτε?
- Δοκιμάστε στο προηγούμενο παράδειγμα να χρησιμοποιήσετε την `getchar`. Τι αλλαγές πρέπει να κάνετε;

```
int x;  
x = getch();  
printf( " \n Ο χαρακτήρας είναι %c " ,x);
```

getchar

- Περιγράφεται στην βιβλιοθήκη `stdio.h` με πρωτότυπο **`int getchar(void)`**
- Διαβάζει ένα χαρακτήρα από την κύρια είσοδο και τον επιστρέφει στο πρόγραμμα.

Άσκηση

Δοκιμάστε στο προηγούμενο παράδειγμα να χρησιμοποιήσετε την `getchar`. Τι αλλαγές πρέπει να κάνετε;

```
int x;  
x = getche();  
printf( " \n Ο χαρακτήρας είναι %c " ,x);
```

Παρατηρήσεις

- οι `getche()` και `getch()` επιστρέφουν τιμές τύπου `int`.
- η `getchar()` επιστρέφει χαρακτήρες
- Είναι απαραίτητο να πατήσετε το πλήκτρο `return` όταν χρησιμοποιείτε τη συνάρτηση `getchar()`.
- Στη `C` οι χαρακτήρες δεν είναι τίποτα άλλο από ακέραιοι αριθμοί. Κάθε χαρακτήρας στη `C` αντιστοιχεί σε έναν ακέραιο αριθμό από 0 έως 254 (γνωστός και ως κωδικός `ASCII`).
- Το σύμβολο `%c` είναι αυτό που λέει στην `printf` ότι θα πρέπει να τυπώσει χαρακτήρα και όχι αριθμό.

putchar

- Ορίζεται στην `stdio.h`
- Δέχεται έναν χαρακτήρα σαν παράμετρο και τον εμφανίζει στην τρέχουσα θέση του δρομέα

Παράδειγμα

```
#include <stdio.h>  
main()  
{  
    char ch;  
  
    ch = getchar();  
  
    putchar(ch);  
}
```

ορίζεται αρχικά μια μεταβλητή τύπου `char` (χαρακτήρα), με το όνομα `ch`.

Ο χαρακτήρας που θα δώσει ο χρήστης από το πληκτρολόγιο θα αποθηκευτεί στη μνήμη στη μεταβλητή `ch`

Ο χαρακτήρας που έχει αποθηκευτεί στη μεταβλητή `ch` επιστρέφεται στην οθόνη

Πίνακας ASCII

Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char	Dec	Char
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	{	56	8	72	H	88	X	104	h	120	x
41	}	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	.	61	=	77	M	93]	109	m	125	}
46	/	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	_
48	0	64	@	80	P	96	`	112	p		

Τελεστές & Ταξινόμηση

- Ταξινόμηση ανάλογα με :
 - A.** τον αριθμό των τελεστών στους οποίους δρουν
 - Μοναδιαίους (unary)
 - Δυαδικούς (binary)
 - Τριαδικούς (ternary)
 - B.** Την διεργασία που εκτελούν
 - Αριθμητικοί + / * -
 - Λογικοί && || !
 - Συσχετιστικοί > >= < <= == !=
 - Διαχείριση bits >> & ! ^
 - Διαχείριση μνήμης& [] . ->

Νέο Πρόγραμμα

- Δημιουργήστε ένα νέο project με τίτλο

Variables2

Άσκηση

Γράψτε ένα πρόγραμμα όπου υπολογίζει τη περίμετρο ενός τετραγώνου με μήκος πλευράς **a**

Άσκηση

επεκτείνετε το πρόγραμμα σας ώστε το a να εισάγεται από το πληκτρολόγιο

Σύμβολα δυαδικών τελεστών

Δυαδικός Τελεστής	Σύμβολο
Μικρότερο	<
Μικρότερο ή ίσο	<=
Ίσο	==
Διάφορο	!=
Μεγαλύτερο	>
Μεγαλύτερο ή ίσο	>=
Πρόσθεση	+
Αφαίρεση	-
Πολλαπλασιασμός	*
Διαίρεση πραγματικών	/
Πηλίκιο διαίρεσης ακεραίων	/
Υπόλοιπο διαίρεσης ακεραίων	%

Νέο Πρόγραμμα

- Δημιουργήστε ένα νέο project με τίτλο

Operators

Τελεστές – παράδειγμα

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

```
int main()
```

```
{
```

```
    int x, y;
```

```
    float z, w;
```

```
    x = 7 / 5;
```

```
    y = 13 % 4;
```

```
    z = 7.0 / 5.0;
```

```
    w = 7 / 5;
```

```
    printf( “ x = %d, y = %d \n ” , x, y );
```

```
    printf( “ z = %f, w = %f \n ” , z, w );
```

```
    return 0;
```

```
}
```

Συμπέρασμα

- **Οι τελεστές συμπεριφέρονται ανάλογα με το είδος των μεταβλητών που παίρνουν μέρος στην πράξη**

Σημειογραφία Τελεστών και C

- Η θέση του τελεστή ανάμεσα σε τελεστέους παρουσιάζει διαφορετική σημειογραφία κάθε φορά.
 - **Ένθετου τελεστή** $a+b$ (μεταξύ τελεστέων)
 - **Προπορευόμενου τελεστή** $+ab$ (πρίν από τελεστέους)
 - **Παρελκόμενου τελεστή** $ab+$. (μετά από τους τελεστέους)

Γλώσσα C – Κατηγορίες Εκφράσεων

- **Σταθερές** (περιέχουν μόνο σταθερές τιμές)
- **Ακέραιες και Κινητής υποδιαστολής** (μετά από άμεσες και έμμεσες μετατροπές τύπων δίνουν αποτέλεσμα τύπου ακεραίου ή κινητής υποδιαστολής)
- **Εκφράσεις Δείκτη** (Εκφράσεις με τιμή μία διεύθυνση. Εκφράσεις με περιεχόμενο μεταβλητές τύπου δείκτη, τελεστή διεύθυνσης (&), αλφαριθμητικές σταθερές και ονόματα πινάκων)

Προτεραιότητα & Προσεταιριστικότητα Τελεστών

- Σχηματισμός ένθετων εκφράσεων με την χρήση παρενθέσεων

$$(((n+5) \leq a) \&\& q)$$

- Η διαδοχική παράθεση τελεστών

- $5*6-4$

$$\rightarrow (5*6)-4 = 26$$

$$\rightarrow 5*(6-4) = 10$$

Προς αποφυγή των παραπάνω δυσχερειών οι τελεστές ταξινομούνται σε επίπεδα προτεραιότητας με τον κανόνα ότι οι τελεστές υψηλού επιπέδου δρουν επί των τελεστών πριν από τους τελεστές χαμηλότερου επιπέδου

Προτεραιότητα Τελεστών

- Η ύπαρξη περισσότερων τελεστών στο ίδιο επίπεδο προτεραιότητας επιβάλλει τον προσδιορισμό κατεύθυνσης εφαρμογής από αριστερά προς τα δεξιά
 - πχ. $8-5-2 = 1$

Προτεραιότητα αριθμητικών τελεστών
()
* / %
+ -

Νέο Πρόγραμμα

- Δημιουργήστε ένα νέο project με τίτλο

Variables3

Άσκηση

Γράψτε ένα πρόγραμμα που να υπολογίζει τον αριθμό ημερών από την αρχή του έτους όταν δίνεται μία ημερομηνία.

Η ημερομηνία αποθηκεύεται στις μεταβλητές:

mera

mhnas

Θεωρήστε ότι όλοι οι μήνες έχουν 30 ημέρες

Προτεραιότητα Τελεστών και Συσχετισμός

Τελεστές	Συσχετισμός
() [] >	Από αριστερά προς τα δεξιά
! ~ ++ -- + - * & (τύπος) sizeof	Από αριστερά προς τα δεξιά
* / % (αριθμητικοί τελεστές)	Από αριστερά προς τα δεξιά
+ - (αριθμητικοί τελεστές)	Από αριστερά προς τα δεξιά
<< >>	Από αριστερά προς τα δεξιά
< <= > >=	Από αριστερά προς τα δεξιά
== !=	Από αριστερά προς τα δεξιά
&	Από αριστερά προς τα δεξιά
^	Από αριστερά προς τα δεξιά
	Από αριστερά προς τα δεξιά
&&	Από αριστερά προς τα δεξιά
	Από αριστερά προς τα δεξιά
?:	Από δεξιά προς τα αριστερά
= += -= *= &= ^= = <<= >>=	Από αριστερά προς τα δεξιά

Νέο Πρόγραμμα

- Δημιουργήστε ένα νέο project με τίτλο

Operators2

Νέο Πρόγραμμα

Υπολογίστε τη περίμετρο και το εμβαδό ενός κύκλου με ακτίνα **R**