Διαδικαστικός Προγραμματισμός

Ενότητα 9: Χαρακτήρες και Αλφαριθμητικά

Καθηγήτρια Μαρία Σατρατζέμη

Τμήμα Εφαρμοσμένης Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

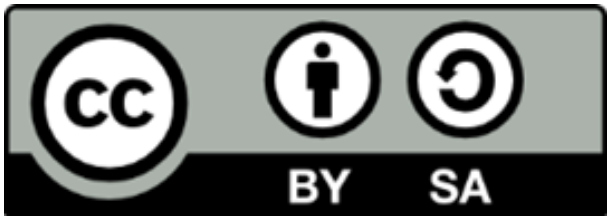
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σκοποί ενότητας

- Να κατανοήσετε τον τρόπο με τον οποίο αναπαριστούν τα συστήματα υπολογιστών τον τύπο δεδομένων char και πώς να χειρίζεστε αντικείμενα αυτού του τύπου.
- Να μάθετε να χρησιμοποιείτε τις συναρτήσεις της διασύνδεσης ctype.h.
- Να κατανοήσετε την έννοια του αφηρημένου τύπου.
- Να μπορείτε να χρησιμοποιείτε τις συναρτήσεις της διασύνδεσης string.h για να χειρίζεστε αλφαριθμητικά.
- Να κατανοήσετε τις αρχές της απαρίθμησης και της κωδικοποίησης με ακεραίους ως στρατηγικές για τον ορισμό νέων τύπων δεδομένων.
- Να είστε σε θέση να ορίζετε και να χειρίζεστε απαριθμητούς τύπους στη C.

Βαθμωτοί τύποι

- Οι τύποι που συμπεριφέρονται ως ακέραιοι ονομάζονται βαθμωτοί τύποι (scalar types).
- Στη C, οι βαθμωτοί τύποι μετατρέπονται αυτομάτως σε ακεραίους κάθε φορά που τους χρησιμοποιούμε σε μια παράσταση.

Χαρακτήρες

- Οι χαρακτήρες αποτελούν τη βάση για την επεξεργασία όλων των δεδομένων κειμένου.
- Αν τα αλφαριθμητικά εμφανίζονται συχνότερα στα προγράμματα, οι μεμονωμένοι χαρακτήρες αποτελούν τον θεμελιώδη τύπο για τη δόμηση όλων των άλλων μορφών δεδομένων κειμένου.
- Οι χαρακτήρες αποτελούν έναν ενσωματωμένο απαριθμητό τύπο (αν και δεν ορίζεται με τη χρήση της λέξης-κλειδιού enum).
- Οι χαρακτήρες είναι ένας βαθμωτός τύπος και συνεπώς συμπεριφέρονται ως ακέραιοι.

Ο τύπος δεδομένων char (1)

- Στη C, οι μεμονωμένοι χαρακτήρες αναπαριστάνονται με τη χρήση του προκαθορισμένου τύπου δεδομένων **char**.
- Πεδίο ορισμού: ανεπίσημα, το πεδίο ορισμού του τύπου δεδομένων **char** είναι το σύνολο των συμβόλων που είναι δυνατό να εμφανιστούν στην οθόνη ή υπάρχουν στο πληκτρολόγιο (γράμματα, ψηφία, σημεία στίξης, το κενό διάστημα, το πλήκτρο Enter κ.ο.κ.).
- Σύνολο πράξεων: εφόσον ο **char** είναι βαθμωτός τύπος, το σύνολο των πράξεων που είναι διαθέσιμες για τους χαρακτήρες είναι το ίδιο με τους ακέραιους.
- Διάβασμα χαρακτήρων από το πληκτρολόγιο: χρησιμοποιείτε τη συνάρτηση **getchar():**

```
char ch;
ch = getchar();
```

- Εμφάνιση χαρακτήρων:

```
printf("%c", ch);
```

Ο τύπος δεδομένων char (2)

- Οι μεμονωμένοι χαρακτήρες αναπαριστάνονται στο εσωτερικό του μηχανήματος, όπως και κάθε άλλος βαθμωτός τύπος, δηλ. με ένα αριθμό.
- Ο κωδικός που χρησιμοποιείται για την αναπαράσταση ενός συγκεκριμένου χαρακτήρα ονομάζεται κωδικός χαρακτήρα (character code).
- Συγκεκριμένα, έχει υιοθετηθεί το γνωστό πρότυπο κωδικοποίησης των χαρακτήρων ASCII.

Ο κώδικας ASCII

	0	1	2	3	4	5	6	7	8	9
0	\000	\001	\002	\003	\004	\005	\006	\a	\b	\t
10	\n	\v	\f	\r	\016	\017	\020	\021	\022	\023
20	\024	\025	\026	\027	\030	\031	\032	\033	\034	\035
30	\036	\037	κενό	!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	\177		

Ιδιότητες της κωδικοποίησης ASCII:

- Οι κωδικοί των χαρακτήρων που παριστάνουν τα ψηφία 0 μέχρι 9 είναι διαδοχικοί.
- Τα γράμματα της αγγλικής αλφαβήτου χωρίζονται σε δύο χωριστά διαστήματα τιμών: ένα για τα κεφαλαία (A-Z) και ένα για τα πεζά γράμματα (a-z). Μέσα σε κάθε διάστημα τιμών, οι τιμές ASCII είναι διαδοχικές.

Ειδικοί χαρακτήρες

Εκτός από τους γνωστούς χαρακτήρες που εμφανίζονται στην οθόνη και ονομάζονται **εκτυπώσιμοι χαρακτήρες** (printing characters), ο πίνακας ASCII περιλαμβάνει και διάφορους **ειδικούς χαρακτήρες** (special characters) που χρησιμοποιούνται για να επιτελούν συγκεκριμένες ενέργειες.

Πίνακας ειδικών χαρακτήρων

\a	ηχητική προειδοποίηση (μπιπ ή κουδούνισμα)
\b	οπισθοδρόμηση (backspace)
\f	αλλαγή σελίδας
\n	αλλαγή γραμμής
\r	επιστροφή (επιστρέφει στην αρχή της τρέχουσας γραμμής χωρίς προώθηση
\t	στηλοθέτης (tab – μεταφέρει στον επόμενο κατακόρυφο στηλοθέτη)
\v	κατακόρυφος στηλοθέτης
\0	μηδενικός (null) χαρακτήρας (ο χαρακτήρας του οποίου ο κωδικός ASCII είναι 0)
****	ο χαρακτήρας \
\'	ο χαρακτήρας ' (απαιτεί ανάποδη κάθετο μόνο σε σταθερές χαρακτήρων)
\”	ο χαρακτήρας “ (απαιτεί ανάποδη κάθετο μόνο σε αλφαριθμητικές σταθερές)
\add	ο χαρακτήρας του οποίου ο κωδικός ASCII είναι ο οκταδικός αριθμός add

Αριθμητική χαρακτήρων (1)

Στη C, οι τιμές χαρακτήρων μπορούν να χρησιμοποιηθούν σαν να ήταν ακέραιοι, χωρίς να απαιτείται κάποια ειδική μετατροπή.

Το αποτέλεσμα μιας τέτοιας πράξης ορίζεται σύμφωνα με τους εσωτερικούς κωδικούς ASCII.

Παρόλο που είναι έγκυρο να εφαρμόζετε οποιαδήποτε αριθμητική πράξη σε τιμές τύπου **char**, δεν έχουν νόημα όλες οι πράξεις για τους χαρακτήρες.

Για παράδειγμα:

η πράξη ‘A’ * ‘B’ είναι έγκυρη

αλλά το αποτέλεσμα 4290 που προκύπτει από τον πολλαπλασιασμό των κωδικών ASCII (65 και 66 αντίστοιχα) βρίσκεται εκτός του συνόλου χαρακτήρων ASCII και δεν έχει καμία σημασία ως χαρακτήρας.

```
int num = 'A' * 'B';
 printf("%d\n", num);     //εκτυπώνει 4290
```

Αριθμητική χαρακτήρων(2)

Πρόσθεση ακεραίου σε χαρακτήρα:

‘**0’ + 5** αποτέλεσμα **‘5’**

Γενικά, η παράσταση **‘0’ + n** δίνει τον κωδικό χαρακτήρα του **n**-οστού ψηφίου, αν το n είναι μεταξύ 0 και 9.

Αφαίρεση ακεραίου από χαρακτήρα:

‘z’ – 2 αποτέλεσμα **‘x’**

Γενικά, η παράσταση **c - n** δίνει τον κωδικό του χαρακτήρα που βρίσκεται **n** χαρακτήρες πριν από το c στην ακολουθία κωδικοποίησης.

Αφαίρεση χαρακτήρα από άλλο χαρακτήρα:

‘a’ – ‘A’ αποτέλεσμα **32**

Γενικά, αν **c1** και **c2** είναι δύο χαρακτήρες τότε η παράσταση **c1 – c2** δίνει την απόσταση μεταξύ αυτών των χαρακτήρων στην ακολουθία κωδικοποίησης.

Σύγκριση δύο χαρακτήρων μεταξύ τους:

Γενικά, η παράσταση **c1 < c2** είναι **TRUE** αν το **c1** εμφανίζεται πριν από το **c2** στον πίνακα ASCII.

Εφαρμογές

Διάγνωση αν ένας χαρακτήρας είναι αριθμητικό ψηφίο:

```
if(ch >= '0' && ch <= '9')
```

Διάγνωση κεφαλαίου αγγλικού γράμματος:

```
if(ch >= 'A' && ch <= 'Z')
```

Μετατροπή πεζού σε κεφαλαίο:

```
char ch;
printf("Dwse ena pezo aggliko gramma: ");
ch = getchar();
printf("To antistoixo kefalaio einai: ");
printf("%c\n", ch-32);
```

Η διασύνδεση ctype.h

islower(ch):	επιστρέφει **TRUE** ο χαρακτήρας στη μεταβλητή **ch** είναι πεζό γράμμα
isupper(ch):	επιστρέφει **TRUE** ο χαρακτήρας στη μεταβλητή **ch** είναι κεφαλαίο γράμμα
isalpha(ch):	επιστρέφει TRUE ο χαρακτήρας στη μεταβλητή ch είναι γράμμα
isdigit(ch):	επιστρέφει TRUE ο χαρακτήρας στη μεταβλητή ch είναι ψηφίο
isalnum(ch):	επιστρέφει TRUE ο χαρακτήρας στη μεταβλητή ch είναι αλφαριθμητικό (γράμμα ή ψηφίο)
ispunct(ch):	επιστρέφει TRUE ο χαρακτήρας στη μεταβλητή ch είναι σημείο στίξης
isspace(ch):	επιστρέφει TRUE ο χαρακτήρας στη μεταβλητή ch είναι ένας από τους εξής: ' ' (κενό διάστημα), '\t', '\n', '\f', '\v' που εμφανίζονται στην οθόνη ως κενό διάστημα
tolower(ch):	επιστρέφει το αντίστοιχο πεζό γράμμα
toupper(ch):	επιστρέφει το αντίστοιχο κεφαλαίο γράμμα

Παράδειγμα

Η παρακάτω κατηγορηματική συνάρτηση επιστρέφει **TRUE** αν το όρισμα της είναι φωνήεν της αγγλικής αλφαβήτου:

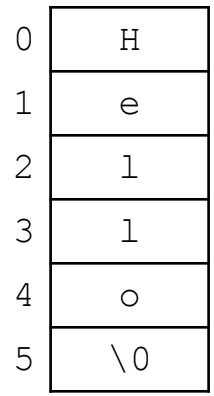
```
bool IsVowel(char ch)
{
    char lowerCh;

    lowerCh = tolower(ch);
  switch( lowerCh )
    {
            case ‘a’:   case ‘e’:   case ‘i’:    case ‘o’:   case ‘u’:
            return (TRUE);
            default:
                        return false;
    }
 }
```

Τα αλφαριθμητικά ως πίνακες (1)

- Στο εσωτερικό του υπολογιστή τα αλφαριθμητικά αναπαριστάνονται ως πίνακες χαρακτήρων.
- Οι χαρακτήρες ενός αλφαριθμητικού αποθηκεύονται σε συνεχόμενα byte.
- Για να προσδιοριστεί που τελειώνει το κάθε αλφαριθμητικό, ο μεταγλωττιστής της C αποθηκεύει πάντα ένα μηδενικό (null) χαρακτήρα **('\0')** στο byte που ακολουθεί αμέσως μετά τον τελευταίο χαρακτήρα ενός αλφαριθμητικού.
- Παράδειγμα: δήλωση του πίνακα **carray** και ανάθεση ως αρχική τιμή του αλφαριθμητικού "**Hello**":

```
char carray[6];
carray[0] = 'H';
carray[1] = 'e';
carray[2] = 'l';
carray[3] = 'l';
carray[4] = 'o';
carray[5] = '\0';
```



Τα αλφαριθμητικά ως πίνακες (2)

- Επειδή κάθε αλφαριθμητικό είναι ένας πίνακας, μπορείτε να επιλέξετε τον ***i***-οστό χαρακτήρα ενός αλφαριθμητικού **str** ως εξής:

  str [i]

- Για να χρησιμοποιήσετε, για οποιοδήποτε λόγο, τους χαρακτήρες ενός αλφαριθμητικού όταν χρησιμοποιείτε συμβολισμό πινάκων, μπορείτε να χρησιμοποιήσετε τον παρακάτω ιδιωματισμό

```
for(i = 0; str[i] != '\0'; i++){
        … σώμα του βρόχου όπου γίνεται ο χειρισμός του str[i]…
}
```

Τα αλφαριθμητικά ως πίνακες (3)

- Στη συνέχεια, παρουσιάζεται η υλοποίηση μιας συνάρτησης που επιστρέφει τον αριθμοδείκτη θέσης του 1ου φωνήεντος της λέξης που δέχεται, ή την τιμή -1 αν η λέξη δεν έχει κανένα φωνήεν

```
int FindFirstVowel(char word[])
{
    int i;
    for(i = 0; word[i] != '\0'; i++){
            if (IsVowel(word[i])) return (i);
    }
    return (-1);
}
```

Η διασύνδεση string.h (1)

Κλήση συνάρτησης	*Επιστρεφόμενη τιμή*
strcpy(dst, src):	αντιγράφει χαρακτήρες από το αλφαριθμητικό **src** στο **dst**
strncpy(dst, src, n):	αντιγράφει το πολύ **n** χαρακτήρες από το αλφαριθμητικό **src** στο **dst**
strcat(dst, src):	προσαρτά χαρακτήρες από το **src** στο τέλος του **dst**
strncat(dst, src, n):	προσαρτά το πολύ **n** χαρακτήρες από το **src** στο τέλος του **dst**
strlen(s):	επιστρέφει το μήκος του αλφαριθμητικού **s**
strcmp(s1, s2):	επιστρέφει έναν ακέραιο που υποδηλώνει το αποτέλεσμα της σύγκρισης των αλφαριθμητικών
strncmp(s1, s2, n):	όπως η, αλλά συγκρίνει το πολύ **n** χαρακτήρες

Συνάρτηση strcopy

Η συνάρτηση strcpy αντιγράφει χαρακτήρες από ένα αλφαριθμητικό (προέλευση) σε ένα άλλο αλφαριθμητικό (προορισμός)

Δέχεται πρώτα το όρισμα του προορισμού

π.χ.

```
char carray[6];
strcpy(carray, "A long string");
```

Το μήκος του αλφαριθμητικού "A long string" είναι μεγαλύτερο από το μέγεθος του πίνακα carray. Αυτό θα προκαλέσει την εγγραφή δεδομένων μετά το τέλος του πίνακα και είναι συνηθισμένο προγραμματιστικό λάθος (υπερχείλιση περιοχής προσωρινής αποθήκευσης - buffer overflow)

Τέτοια σφάλματα μπορεί να αποδειχθούν εξαιρετικά κρίσιμα σε θέματα ασφάλειας.

Μεταβίβαση αλφαριθμητικών σε συναρτήσεις

Τα αλφαριθμητικά μπορούν να μεταβιβαστούν σε συναρτήσεις ακριβώς όπως και οποιοσδήποτε άλλος πίνακας (κλήση κατ' αναφορά)

Παράδειγμα: συνάρτηση αντιστροφής των χαρακτήρων ενός αλφαριθμητικού

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

void reverse(char array[]);

main() {
  char string[1000] = "Hello my friend";
  reverse(string);
  printf("%s\n", string);

  system("PAUSE");
}
```

```
void reverse(char array[]) {

  char temp[1000];
  int i, end;

  strcpy(temp, array);  //αντιγραφή του αλφαριθμητικού
                        //στον προσωρινό πίνακα temp

  for(i=0; array[i] != '\0'; i++)  //εύρεση τέλους
  end = i;

  i = 0;
  while(end >=0) {
    array[i] = temp[end];
    i++;
    end--;
  }
  array[i] = '\0';
}
```

Αλφαριθμητικά

1. char name1[20], *name2; 2 τρόποι για να δηλώσω αλφαριθμητικό

2. printf("DWSE 1o ONOMA ");
3. gets(name1); ← Με την gets() διαβάζω αλφαριθμητικό
4. printf("%s\n",name1);
5.
6. printf("DWSE 2o ONOMA ");
7. // gets(name2); runtime error
8. name2=(char *)malloc(20*sizeof(char));
9. gets(name2);

Αν δηλώσω το αλφαριθμητικό ως δείκτη προς χαρακτήρα τότε όπως οι εντολές < 8. 9.> για να διαβάσω το αλφαριθμητικό

10. // name1=name2; error: incompatible types in assignment
•
11. strcpy(name1, name2);

Εντολή ανάθεσης τιμής δε μπορώ να χρησιμοποιήσω μεταξύ αλφαριθμητικών αλλά την strcpy(dst, src)

Αλφαριθμητικά ως Δείκτες

Όπως συμβαίνει με οποιονδήποτε πίνακα, ένας πίνακας χαρακτήρων μπορεί να ερμηνευτεί και ως δείκτης προς το πρώτο του στοιχείο.

Επομένως, μέσω αριθμητικής δεικτών μπορεί ο προγραμματιστής να προσπελάσει μεμονωμένους χαρακτήρες ενός αλφαριθμητικού.

Παράδειγμα εύρεσης πρώτου φωνήεντος σε αλφαριθμητικό

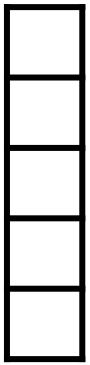
```
int FindFirstVowel(char* word) {
  char *cp;

  for(cp = word; *cp != '\0'; cp++)
    if(IsVowel(*cp))
      return cp-word;
  return (-1);
}
```

Η αρχή της απαρίθμησης (1)

- Σε αρκετές περιπτώσεις χρειάζεται να γράφουμε προγράμματα που χειρίζονται μη αριθμητικά δεδομένα.
- Για παράδειγμα, σε ένα πρόγραμμα υποβολής φορολογικής δήλωσης υπάρχει μια ερώτηση του τύπου

Οικογενειακή κατάσταση (επιλέξτε ένα):



ανύπαντρος

παντρεμένος που υποβάλει κοινή δήλωση

παντρεμένος που υποβάλει χωριστή δήλωση

επικεφαλής νοικοκυριού

χήρος/χήρα

- Η απάντηση σας δεν είναι ούτε αριθμητικά δεδομένα ούτε δεδομένα κειμένου.

Η αρχή της απαρίθμησης (2)

- Ο καλύτερος τρόπος για να περιγράψετε τον τύπο δεδομένων θα ήταν απλώς να τον ονομάσετε ***δεδομένα οικογενειακής κατάστασης***.
- Αυτός ο νέος τύπος δεδομένων έχει ως *πεδίο ορισμού* 5 τιμές: **ανύπαντρος**, **παντρεμένος που υποβάλει κοινή δήλωση, παντρεμένος που υποβάλει χωριστή δήλωση, επικεφαλής νοικοκυριού, χήρος/χήρα.**
- Η διεργασία της παράθεσης όλων των δεδομένων του πεδίου ορισμού ενός τύπου δεδομένων με τη μορφή λίστας ονομάζεται απαρίθμηση (enumeration).
- Ένας τύπος που ορίζεται με την παράθεση όλων των στοιχείων του ονομάζεται απαριθμητός τύπος (enumeration type).

Αναπαράσταση απαριθμητών τύπων

- Σε κάθε στοιχείο ενός απαριθμητού τύπου μπορούμε να αναθέσουμε ένα ακέραιο, διεργασία γνωστή ως **κωδικοποίηση με ακεραίους** (integer encoding).
- Για το παράδειγμα μας μπορούμε να εφαρμόσουμε αυτή τη στρατηγική ως εξής:

```
#define Single                          1
#define MarriedFillingJointReturn       2
#define MarriedFillingSeperateReturn    3
#define HeadOfHousehold                 4
#define QualifyingSurvivingSpouse       5
```

Από τη στιγμή που θα ορίσετε τις σταθερές μπορείτε να δηλώσετε μια μεταβλητή τύπου **int** για να αναπαραστήσετε την οικογενειακή κατάσταση:

```
int filingStatus;
```

Ορισμός νέων απαριθμητών τύπων

- Στη C, αντί να χρησιμοποιήσετε τον τύπο **int** για να αναπαραστήσετε μη αριθμητικές τιμές, είναι δυνατό να ορίσετε ένα πραγματικό όνομα τύπου ο οποίος θα αναπαριστά έναν απαριθμητό τύπο.

typedef enum{
λίστα στοιχείων
} όνομα τύπου;

όπου***:***
- ***λίστα στοιχείων*** είναι ένας κατάλογος ονομάτων τα οποία είναι δυνατό να χρησιμοποιηθούν για την αναφορά στις μεμονωμένες τιμές που απαρτίζουν τον απαριθμητό τύπο. Τα στοιχεία της λίστας διαχωρίζονται με κόμματα. Κάθε στοιχείο μπορεί επίσης να ακολουθείται από ένα σύμβολο ίσον και μια ακέραιη σταθερά που καθορίζει μια συγκεκριμένη εσωτερική αναπαράσταση
- ***όνομα_τύπου*** είναι το όνομα του νέου απαριθμητού τύπου

Παράδειγμα (1)

- Ορισμός ενός νέου τύπου **weekdayT** με πεδίο ορισμού τα ονόματα των ημερών της εβδομάδας

```
typedef enum{
        Sunday, Monday, Tuesday, Wednesday,
        Thursday, Friday, Saturday
} weekdayT
```

- Στα στοιχεία του παραπάνω απαριθμητού τύπου ανατίθενται συνεχόμενες ακέραιες τιμές ξεκινώντας από το 0.

- Βέβαια, όπως θα δούμε στο επόμενο παράδειγμα, ο μεταγλωττιστής της C σας επιτρέπει να καθορίσετε ρητά την εσωτερική αναπαράσταση των στοιχείων ενός απαριθμητού τύπου κατά τον ορισμό του.

Παράδειγμα (2)

- **Για το παράδειγμα με τη φορολογική δήλωση μπορούμε να έχουμε τη δήλωση του απαριθμητού τύπου**

```
typedef enum{
        Single = 1,
        MarriedFillingJointReturn = 2,
        MarriedFillingSeperateReturn = 3,
        HeadOfHousehold = 4,
        QualifyingSurvivingSpouse = 5
} filingStatusT;
```

ή

```
typedef enum{
        Single = 1,
        MarriedFillingJointReturn,
        MarriedFillingSeperateReturn,
        HeadOfHousehold,
        QualifyingSurvivingSpouse
} filingStatusT;
```

Πράξεις με απαριθμητούς τύπους

```
typedef enum{
                Sunday, Monday, Tuesday, Wednesday,
                Thursday, Friday, Saturday
} weekdayT
```

- Αφού οι απαριθμητοί τύποι είναι στη ουσία ακέραιοι μπορούμε να κάνουμε πράξεις, όπως για παράδειγμα

  weekday = weekday + 1;

- Πρέπει όμως να είμαστε προσεκτικοί γιατί ο μεταγλωττιστής δεν ελέγχει αν το αποτέλεσμα είναι έγκυρο μέλος ενός συγκεκριμένου απαριθμητού τύπου. Αν γράφαμε την παραπάνω εντολή ανάθεσης και η **weekday** είχε την τιμή **Saturday**, ο υπολογιστής θα έπαιρνε την τιμή 6 θα πρόσθετε το 1 και θα αποθήκευε την τιμή 7 στη μεταβλητή **weekday.**
- Αντίθετα, η εντολή

  weekday = (weekday + 1) % 7;

- εξασφαλίζει ότι το αποτέλεσμα θα είναι πάντα από 0 έως 6.

Τέλος Ενότητας

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης