

ΑΛΦΑΡΙΘΜΗΤΙΚΑ

1. Χρήση χαρακτήρων και αλφαριθμητικών στη C

1.1 Ο Τύπος Δεδομένων char

Ομοίως με τους άλλους τύπους μεταβλητών που έχουν αναφερθεί στα προηγούμενα εργαστήρια, ο τύπος char περιλαμβάνει σταθερές και μεταβλητές.

1.1.1 Σταθερές Τύπου char

Μια σταθερά τύπου char είναι ένας χαρακτήρας που εσωκλείεται σε μονά εισαγωγικά (π.χ. 'α', 'x', '^') και ουσιαστικά αντιπροσωπεύει έναν αριθμό, όπως αυτός καθορίζεται από το σύστημα ASCII. Η παρουσία επομένως τέτοιων χαρακτήρων εντός αριθμητικών παραστάσεων δεν αποτελεί συντακτικό ή λογικό λάθος για τα δεδομένα της 'C'. Ακολουθεί κατατοπιστικό παράδειγμα.

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>
main ()
{
    printf(" o xarakthras %c antistoixei ston arithmo %d\n", 'P', 'P' );
    printf(" o xarakthras %c antistoixei ston arithmo %d\n", 'p', 'p' );
    printf(" o xarakthras %c antistoixei ston arithmo %d\n", 'b', 'b' );
    printf(" h parastash b + p - 187 isodynamei me ton arithmo %d\n", 'b' + 'p' - 187 );
    return 0;
}
```

Με τα εξής αποτελέσματα :

```
o xarakthras P antistoixei ston arithmo 80
o xarakthras p antistoixei ston arithmo 112
o xarakthras b antistoixei ston arithmo 98
h parastash b + p - 187 isodynamei me ton arithmo 23
-----
Process exited after 0.01114 seconds with return value 0
Press any key to continue . . .
```

1.1.2 Μεταβλητές Τύπου char

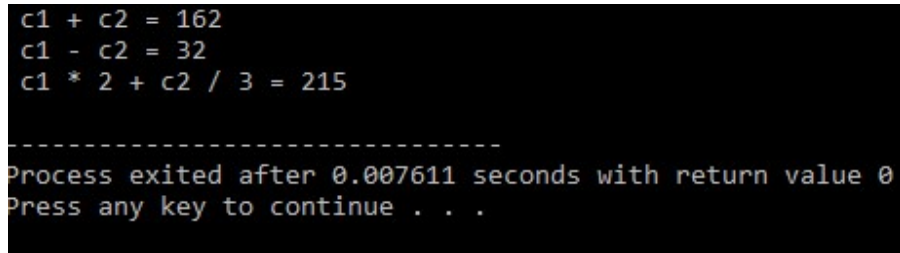
Μια μεταβλητή τύπου char δεσμεύει 1 byte στην μνήμη του συστήματος μας. Αντίστοιχα με τις σταθερές που προηγήθηκαν στην παραπάνω υποπαράγραφο, ο ASCII κωδικός του εκάστοτε χαρακτήρα αποθηκεύεται στην αντίστοιχη μεταβλητή. Αρκεί φυσικά να έχει προηγηθεί η σωστή δήλωση στο πεδίο ορισμού μεταβλητών. Ακολουθεί κατάλληλο παράδειγμα με τη δήλωση μεταβλητών τύπου char και αριθμητικές πράξεις με αυτές:

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>
main ()
{
    int x , y;
    char c1 = 'a', c2 = 'A';

    printf(" c1 + c2 = %d\n", c1 + c2 );
    printf(" c1 - c2 = %d\n", c1 - c2 );
    printf(" c1 * 2 + c2 / 3 = %d\n", c1 * 2 + c2 / 3 );
    return 0;
}
```

Με τα εξής αποτελέσματα :



```
c1 + c2 = 162
c1 - c2 = 32
c1 * 2 + c2 / 3 = 215
-----
Process exited after 0.007611 seconds with return value 0
Press any key to continue . . .
```

1.2 Ανάγνωση / Εμφάνιση Χαρακτήρων

Για την εξυπηρέτηση της εισόδου από το πληκτρολόγιο και της εξόδου μέσω της οθόνης ενός και μόνο χαρακτήρα κάθε φορά, χρησιμοποιούνται οι συναρτήσεις *getch()* και *putch()*. Η μεν πρώτη περιμένει να πληκτρολογηθεί ένας χαρακτήρας και επιστρέφει τον ASCII κωδικό αυτού, χωρίς να πατηθεί το <Enter>, ενώ η δεύτερη εμφανίζει στην οθόνη τον χαρακτήρα που έχει αποθηκευτεί σε μία μεταβλητή char ή αν δοθεί σαν όρισμα εισόδου ένας ακέραιος, τον χαρακτήρα που αντιστοιχεί σε αυτόν. Παρατηρήστε στο επόμενο παράδειγμα πως δεν εισάγεται αυτόματα κάποιος χαρακτήρα διαφυγής. Αντιθέτως, τα αποτελέσματα της *putch* τυπώνονται στο σημείο που είναι τοποθετημένος ο κέρσορας.

ΠΑΡΑΔΕΙΓΜΑ

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

main()
{
    char c;
    printf("dose enan xarakthra: \n");
    c = getch();
    putch(c);
    putch('\n');
    printf("the ASCII code of character %c is %d \n" , c, c );
    printf("\n");
    putch(99);
    putch(23);
    putch(67);
    printf("\n");
}
```

Με τα εξής αποτελέσματα :

```
dose enan xarakтира:  
e  
the ASCII code of character e is 101  
  
c␣  
Press any key to continue . . .
```

1.3 Ιδιότητες Αλφαριθμητικών

Ως αλφαριθμητικό (character strings) θα θεωρούμε στη Γλώσσα Προγραμματισμού 'C' κάθε σύνολο χαρακτήρων που βρίσκεται εντός διπλών εισαγωγικών (π.χ. "cied power", "F16", " ", "Am12rt!") . Ορισμένες παραπλήσιοι ορισμοί είναι οι εξής: συμβολοσειρά, σύνολο χαρακτήρων ή σειρά χαρακτήρων και απλούστερα από την αγγλική ορολογία strings. Μία πολύ σημαντική ιδιότητα, ιδιαίτερα κατά τη χρήση των pointers με τα αλφαριθμητικά, είναι η αποθήκευση του κάθε χαρακτήρα του εκάστοτε αλφαριθμητικού σε συνεχόμενες θέσεις μνήμης. Η αρχική θέση δεν μας αφορά από πού θα ξεκινήσει – μιας και αποτελεί τυχαίο γεγονός – αλλά ο μεταγλωττιστής της 'C' εισάγει τον χαρακτήρα '\0' (null) στο τέλος κάθε αλφαριθμητικού.

Κατά την αποθήκευση λοιπόν των αλφαριθμητικών, δεσμεύεται ένα σύνολο συνεχόμενων θέσεων μνήμης. Επειδή όμως κάθε παράσταση στη 'C' αναπαρίσταται από μία και μόνο τιμή, έτσι και τα αλφαριθμητικά θα πρέπει να αντιστοιχηθούν με μία μονοσήμαντη τιμή. Για την εξυπηρέτηση αυτού του σκοπού, η διεύθυνση του πρώτου χαρακτήρα του εκάστοτε αλφαριθμητικού λειτουργεί ως η τιμή που αντιστοιχίζεται σε ολόκληρο το αλφαριθμητικό. Αυτό το γεγονός φυσικά μπορεί να οδηγήσει σε φαινομενικά περίεργα αποτελέσματα, όπως η σύγκριση δύο ολόιδιων αλφαριθμητικών: "abc" == "abc". Η προηγούμενη λογική συνθήκη επιστρέφει False, μιας και το κάθε ένα από αυτά τα αλφαριθμητικά έχει αποθηκευτεί σε διαφορετική θέση μνήμης.

1.4 Χειρισμός Αλφαριθμητικών μέσω Πινάκων

Γνωρίζοντας τόσο το σημείο έναρξης ενός αλφαριθμητικού (διεύθυνση πρώτου στοιχείου, αλλά και η τιμή ολόκληρου του αλφαριθμητικού) όσο και το τελευταίο χαρακτήρα αυτού (αναζητώντας απλά τον χαρακτήρα '\0') μπορούμε να επεξεργαστούμε ποικιλοτρόπως οποιαδήποτε σειρά χαρακτήρων. Φυσικά αυτές οι διαδικασίες προϋποθέτουν την μεταβίβαση του αλφαριθμητικού σε μία μεταβλητή. Επειδή στη 'C' δεν υπάρχει κάποιος τύπος μεταβλητής για σύνολα χαρακτήρων, παρά μόνο για μεμονωμένους χαρακτήρες, η διαδικασία αποθήκευσης ενός αλφαριθμητικού εξυπηρετείται μέσω των μονοδιάστατων πινάκων χαρακτήρων. Επομένως, ορισμένες παραδείγματα αρχικοποίησης αλφαριθμητικών είναι τα εξής:

```
char score[20];  
char password[10] = "J2F_bong";
```

Προσοχή στο γεγονός πως απαιτείται να δηλώσουμε τουλάχιστον μία θέση περισσότερη από το πλήθος των χαρακτήρων του αλφαριθμητικού που σκοπεύουμε να εισάγουμε, μιας και τοποθετείται αυτόματα από τον μεταγλωττιστή στην τελευταία θέση ο χαρακτήρας null. Πιο συγκεκριμένα, η είσοδος αλφαριθμητικών πραγματοποιείται με τη συνάρτηση *gets()* ενώ η εμφάνιση με την *puts()*. Η μεν πρώτη αναμένει την πληκτρολόγηση ενός συνόλου χαρακτήρων από το χρήστη και με το πάτημα της εντολής <Enter> αποθηκεύει σειριακά τους χαρακτήρες που δόθηκαν ξεκινώντας από τη διεύθυνση που ορίζει το όρισμα της. Αντίστοιχα, η δεύτερη κατά σειρά συνάρτηση είναι αρμόδια για την εμφάνιση ενός αλφαριθμητικού στην οθόνη μας. Κατά την ολοκλήρωση της εκτέλεσης αυτής της εντολής τοποθετείται αυτόματα ο χαρακτήρας διαφυγής που επιτρέπει

αλλαγής γραμμής. Επιπλέον, κατά την επιτυχή εκτέλεση της puts() επιστρέφεται η τιμή 0 ενώ μία τυχαία τιμή διάφορη του μηδενός στην αντίθετη περίπτωση.

Εναλλακτικά, μία μεταβλητή τύπου δείκτη μπορεί επίσης να υποστηρίξει την αποθήκευση ενός αλφαριθμητικού, μιας και η ίδια δείχνει σε μία συγκεκριμένη διεύθυνση, ότι ακριβώς δηλαδή προσφέρει και η πρώτη θέση ενός μονοδιάστατου πίνακα. Ακολουθούν κατατοπιστικά παραδείγματα:

ΠΑΡΑΔΕΙΓΜΑ

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

main()
{
    char msg[20], *words;
    char *t;

    printf("give a phrase: \n");
    t = gets(msg);
    putchar('\n');

    printf("1) the specific place id of phrase %s is %p \n" , msg, msg );
    printf("2) the gets function also returns %p\n" , t);
    puts("");

    words = "alternative way of defining a string";
    puts(words);
}
```

Με τα εξής αποτελέσματα :

```
give a phrase:
This is a simple example

1) the specific place id of phrase This is a simple example is 000000000022FE20
2) the gets function also returns 000000000022FE20
alternative way of defining a string
```

ΠΑΡΑΔΕΙΓΜΑ

```
/* αφαίρεση κενών από ένα τυχαίο αλφαριθμητικό που εισάγει ο χρήστης και μέτρηση των συνολικών
κενών */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

main()
{
    char lex[40];
    int i=0,j,c=0;
    printf("Dose mia frash:");
    gets(lex);
```

```
while (lex[i]!='\0')
{
    if(lex[i]==' ')
    {
        j=i;
        while(lex[j]!='\0')
        {
            lex[j]=lex[j+1];
            j++;
        }

        c++;
        i--;
        printf(" %d) meta thn afairesh tou %dou kenou h frash s einai: %s \n" , c, c, lex);
    }
    i++;
}
puts(lex);
printf("Afairethhkan %d kena\n",c);
}
```

Με τα εξής αποτελέσματα :

```
Dwse mia frash: which programming language is the best?
1) meta thn afairesh tou 1ou kenou h frash s einai: whichprogramming language is the best?
2) meta thn afairesh tou 2ou kenou h frash s einai: whichprogramminglanguage is the best?
3) meta thn afairesh tou 3ou kenou h frash s einai: whichprogramminglanguageis the best?
4) meta thn afairesh tou 4ou kenou h frash s einai: whichprogramminglanguageisthe best?
5) meta thn afairesh tou 5ou kenou h frash s einai: whichprogramminglanguageisthebest?
whichprogramminglanguageisthebest?
Afairethhkan 5 kena
Press any key to continue . . .
```

1.5 Χρήσιμες Συναρτήσεις για Χειρισμό Αλφαριθμητικών

Ακολουθεί ένα συνοπτικό παράδειγμα χρήσης των 5 βασικών συναρτήσεων για το χειρισμό αλφαριθμητικών που προσφέρονται έτοιμες μέσω της 'C' (strlen, strcmp, strcpy, strcat, strstr). Κατά σειρά, η κάθε μία από αυτές πραγματοποιεί την εξής λειτουργία: επιστρέφει έναν ακέραιο που ισούται με το μήκος του εξεταζόμενου αλφαριθμητικού, επιστρέφει έναν ακέραιο από τους -1, 0 και 1 ανάλογα με το αν το πρώτο εκ των δύο συγκρινόμενων αλφαριθμητικών είναι μικρότερο, ίσο ή μεγαλύτερο **αλφαβητικά**, αντιγραφή του δεύτερου αλφαριθμητικού στο πρώτο, προσθήκη ενός αντιγράφου του δεύτερου αλφαριθμητικού στο τέλος του πρώτου και επιστροφή του δείκτη που βρέθηκε το δεύτερο αλφαριθμητικό εντός του πρώτου εκτός αν δε βρεθεί, οπότε επιστρέφει ο χαρακτήρας null. Ακολουθεί ενδεικτικό παράδειγμα:

ΠΑΡΑΔΕΙΓΜΑ

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

main()
{
    char *w1 , *w2 , *w, words[20] = " a simple " , *where;
    int length, k;
```

Πανεπιστήμιο Πελοποννήσου – Σχολή Μηχανικών , Τμήμα ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Διαδικασιακός Προγραμματισμός (Εργαστήριο)

```
w1 = "biggest";
w2 = "smallest";

length = strlen(w1);
printf(" the word %s has length %d \n" , w1 , length);

k = strcmp(w1,w2);
printf("the result of strcmp between %s and %s is %d \n", w1, w2, k);

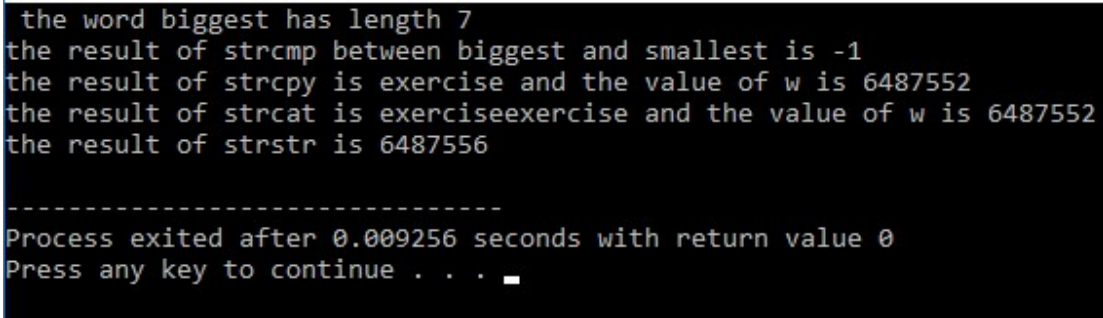
w = strcpy(words,"exercise");
printf("the result of strcpy is %s and the value of w is %d\n", words, w );

w = strcat(words,"exercise");
printf("the result of strcat is %s and the value of w is %d\n", words, w );

where = strstr(words,"cise");
printf("the result of strstr is %d\n", where );

return 0;
}
```

Με τα εξής αποτελέσματα :



```
the word biggest has length 7
the result of strcmp between biggest and smallest is -1
the result of strcpy is exercise and the value of w is 6487552
the result of strcat is exercisexercis and the value of w is 6487552
the result of strstr is 6487556

-----
Process exited after 0.009256 seconds with return value 0
Press any key to continue . . . █
```