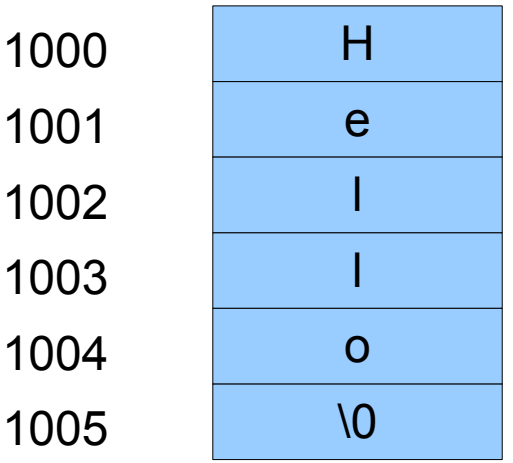
Αλφαριθμητικά, πίνακες και δείκτες

Τα αλφαριθμητικά ως πίνακες

- Ένα αλφαριθμητικό (string) αναπαρίσταται εσωτερικά ως ένας πίνακας χαρακτήρων που τερματίζεται από τον ειδικό χαρακτήρα NULL ('\0')



“Hello”

Τα αλφαριθμητικά ως πίνακες (συνέχ).

- Το παραπάνω δημιουργείται με τις εξής εντολές:

```
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 (συνέχ.)

- Το ίδιο αποτέλεσμα έχουν και οι εντολές

```
char carray[] = “Hello”;

char carray[6] = "Hello";

char carray[] = {'H','e','l','l','o','\0'};

char carray[6] = {'H','e','l','l','o','\0'};
```

Τα αλφαριθμητικά ως πίνακες (συνέχ.)

- Για την προσπέλαση κάθε χαρακτήρα ενός αλφαριθμητικού:

```
int i;
char s[] = "Hello, John";
for (i=0; s[i] != '\0';i++)
{
   κάνε κάτι με το στοιχείο s[i]
}
```

Παράδειγμα: Η συνάρτηση FindFirstVowel

- Με χρήση αφηρημένων αλφαριθμητικών (βιβλιοθήκη strlib)

```
int FindFirstVowel(string word)
{
   int i;
   for (i=0; i<StringLength(word); i++) {
      if (IsVowel(IthChar(word,i)) return (i);
   }
   return (-1);
}
```

Παράδειγμα: Η συνάρτηση FindFirstVowel

- Με χρήση πινάκων

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

Παράδειγμα: Η συνάρτηση FindFirstVowel

- Με χρήση δεικτών

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

Τα αλφαριθμητικά ως αφηρημένος τύπος

Η δήλωση

```
char *s;
```

είναι ισοδύναμη με την

```
string s;
```

Στη βιβλιοθήκη genlib.h υπάρχει ο ακόλουθος ορισμός

```
typedef char *string;
```

Δείκτες και μεταβλητές πίνακα για αλφαριθμητικά

- Το παρακάτω τμήμα κώδικα είναι έγκυρο

```
char *carray;
carray = “A fool on the hill”;
```

- Το παρακάτω τμήμα κώδικα δεν θα μεταγλωττιστεί

```
char carray[32];
carray = “Another Green World”;
```

- Θα προκύψει ένα μήνυμα σφάλματος στη μεταγλώττιση της μορφής “Lvalue required...”

Η βιβλιοθήκη strlib.h (ANSI C)

- strcpy(dst, src)
- strncpy(dst, src, n)
- strcat(dist, src)
- strncat(dist, src,n)
- strlen(s)
- strcmp(s1,s2)
- strchr(s,ch)
- strrchr(s, ch)
- strstr(strpattern,string)

Η συνάρτηση strcpy

strcpy(char dst[], char src[])

Αντιγράφει τη συμβολοσειρά src στην συμβολοσειρά dst.

Παράδειγμα

```
char buffer[128];
strcpy(buffer,”Elearnor Rigby”);
```

Μια υλοποίηση της strcpy

```
void strcpy(char dst[], char src[])
{
  int i;
   for (i = 0; src[i] != '\0'; i++) {
      dst[i] = src[i];
   }
   dst[i] = '\0';
}
```

Μια “παραδοσιακή” υλοποίηση της strcpy

```
void strcpy(char *dst, char *src)
{
    while (*dst++ = *src++);
}
```

Υπερχείλιση

- Με τη χρήση της strcpy είναι δυνατόν να συμβεί υπερχείλιση περιοχής προσωρινής αποθήκευσης (buffer overflow)

```
char carray[6];

strcpy(carray, “This is a long string”);
```

Η συνάρτηση strcat

strcat (char dest[], char src[])

Αντιγράφει τη συμβολοσειρά dest στο τέλος της συμβολοσειράς src

Παράδειγμα

```
char name[128];
strcpy(name, “John”);
strcat(name, “ ”);
strcat(name, “Lennon”);
```

Η συνάρτηση strncat

strncat (char dest[], char src[], int n)

Αντιγράφει n το πολύ χαρακτήρες της συμβολοσειράς dest στο τέλος της συμβολοσειράς src

Παράδειγμα

```
char name[128];
strcpy(name, “John”);
strcat(name, “ ”);
strncat(name, “Lennon”, 3); /* “John Len” */
```

Συναρτήσεις αναζήτησης

char * strchr(char s[], char ch)

Επιστρέφει έναν δείκτη προς τη θέση σε ένα αλφαριθμητικό όπου βρέθηκε ο χαρακτήρας ch

char * strrchr(char s[], char ch)

Ίδια λειτουργία με την strchr ξεκινώντας από το τέλος του αλφαριθμητικού

char * strstr(char s1, char s2[])

Αναζητά την πρώτη εμφάνιση της συμβολοσειράς s2 στην s1.

Εφαρμογή 1

- Να φτιαχτεί συνάρτηση η οποία δέχεται ως είσοδο μια συμβολοσειρά και επιστρέφει τη **θέση του μεγαλύτερου στη σειρά χαρακτήρα** της συμβολοσειράς, σύμφωνα με τη διάταξη των χαρακτήρων

'a', 'b', ...,'x','y',z', 'A', 'B', ..., 'X', 'Y', 'Z',...

Παράδειγμα

για τη συμβολοσειρά “another” επιστρέφεται το 3, η θέση του 't'.

Εφαρμογή 2

- (Άσκηση 1 σελ 600). Να φτιαχτεί μια συνάρτηση η οποία δέχεται ως όρισμα ένα αλφαριθμητικό ως πίνακα χαρακτήρων και επιστρέφει το μέγεθος του αλφαριθμητικού. Πρόκειται για εναλλακτική υλοποίηση της **`strlen`**.
-
- Να γίνει με for και με while

Η βιβλιοθήκη ctype (επιλογή)

- **`isalnum,isalpha`** Ελέγχει αν ένας χαρακτήρας είναι αλφαριθμητικός (A-Z, a-z, 0-9). *Επιστρέφει 1 σε περίπτωση αν ο χαρακτήρας είναι αλφαριθμητικός και 0 αλλιώς*
- **`isdigit`** Ελέγχει αν ένας χαρακτήρας είναι ψηφίο (0-9)
- **`islower`** Ελέγχει αν ένας χαρακτήρας είναι μικρό (λατινικό) γράμμα (a-z).
- **`isupper`** Ελέγχει αν ένας χαρακτήρας είναι κεφαλαίο (λατινικό) γράμμα (A-Z).

Ένα σύνθετο παράδειγμα

- Πρόγραμμα invert.c σελ. 594.
- Πρόγραμμα αντιστροφής ενός ονόματος.
- Παράδείγματα:
  - Δίνοντας το όνομα “Andreas F. Papasalouros” παίρνουμε το “Papasalouros, Andreas F.”
  - Δίνοντας “Steve Miller” παίρνουμε “Miller, Steve”