CODE ACADEMY

ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ

СРЕЩА 10 Низове и указатели

CODE ACADEMY

Масиви и указатели

```
int arr[5];

int * ptr = arr;          // Integer pointer pointing at arr[0]

ptr[0] = 10;              // Assigns 10 to arr[0]

ptr++;                    // ptr now points at arr[1]

ptr--;                    // ptr now points back at arr[0]

*(ptr + 4) = 100;         // Assigns 100 to arr[4].

                          // Note, ptr currently pointing at arr[0] not arr[1].

                          // Hence (ptr + 4) will point at arr[4]

*(arr + 0) = 10;          // Assigns 10 to arr[0]
```

CODE ACADEMY

Масиви и указатели

Името на масива се интерпретира по време на компилация
като указател към нулевия елемент от масива.

```
int main()
{
    int arr[] = {10, 20, 30, 40, 50};  // Integer array
    int *ptr = arr;  // Pointer to 0th element of array
    /*
     * If arr behaves as a constant pointer then compiler
     * must complain about arr++. Since arr++ is equivalent to
     * arr = arr + 1 which is not permitted.
     */
     arr++; // Error
     // No error
     ptr++;

    return 0;
}
```

CODE ACADEMY

Примери:

```
int arr[] = {10, 20, 30, 40, 50};

arr[0];  // Equivalent to *(arr + 0)
arr[4];  // Equivalent to *(arr + 4)

arr[0]          =>  *(arr + 0)
*(arr + 0)      =>  *(0 + arr)  // Since additions are associative

*(arr + 0)      =>  arr[0]
```

CODE ACADEMY

Символи в езика C

Типът Character съдържа

`'a'`, `'b'`, `'c'`, ...`'z'`, `'0'`, `'1'`, ... `'9'`, `'+'`, `'-'`, `'='`, `'!'`, `'~'`, etc, `'\n'`, `'\t'`, `'\0'`, etc.

A-Z, a-z, 0-9 са подредени, така, че може да се извършва аритметика

`char` е еднобитов тир данни способен да съхранява символ

Третира се като аритметичен цял тип данни, обикновено без знак **`unsigned`**

CODE ACADEMY

Символи в езика *C*

Дефиниция:– *string* е масив от символи, завършващ с `'\0'`

```
char t[] = "This is an initialized string!";

char *u = "This is another string!";
```

String се ограничава с двойни кавички

Може да съдържа произволни символи, включително `\"` и `\'`

String константите могат да обхващат и няколко реда

`"Hello, " "World!\n"` е едно и също с `"Hello, World!\n"`

CODE ACADEMY

Символи в езика C

Нека `char *u = "This is another string!";`

Тогава

```
u[0] == 'T'
u[1] == 'h'
u[2] == 'i'

...
u[21] == 'g'
u[22] == '!'
u[23] == '\0'
```

CODE ACADEMY

Символи в езика C

Повечето манипулации със стринг се правят през библиотеката `<string.h>`

String функциите зависят от константата `'\0'`

Така, ме не е нужно да се проят знаците

Примери:

`int strlen(char *s)` – връща дължината на стринг

Като се изключи литерала `'\0'`

`char* strcpy(char *s, char *ct)` – копира стринга `ct` във стринга `s`, като връща `s`

Вградени функции от string.h

`int strcmp(char *s, char *t)`-сравнява лексикографски `s` и `t`, връща

число <0 ако `s < t`,
число >0 ако `s > t`,
нула, ако `s` и `t` са идентични

`char* strcat(char *s, char *ct)`

Конкатенира string `ct` към края на string `s`, връща като резултат `s`

`s` трябва да е достатъчно голямо, за да може да съдържа съдържанието и на двата стринга

Вградени функции от string.h

```
char *strchr(char *str, int ch);
```

намира първото появяване на определен символ в низ.

```
char *strrchr(char *str, int ch);
```

намира последното появяване на определен символ в низ.

```
size_t strcspn(char *str1, char *str2);
```

търси един низ за първото появяване на някой от символи във втори низ

Функцията strcspn() започва да търси от първия знак на str1, като търси някой от отделните символи, съдържащи се в str2.

Функцията не търси низа str2, а само символите, които съдържа.

Ако функцията намери съвпадение, тя връща отместването от началото на str1, където се намира съответстващият символ.

Ако не намери съвпадение, strcspn() връща стойността на strlen(str1).

Това показва, че първото съвпадение е нулевият символ, с който завършва низа

Вградени функции от string.h

```
char *strpbrk(char *str1, char *str2);
```

търсене в един низ за първото появяване на всеки знак, съдържащ се в друг низ.
Различава се по това, че не включва завършващите нулеви знаци в търсенето.

```
char *strstr(char *str1, char *str2);
```

Тази функция търси първото появяване на един символен низ в друг и търси целия string, а не отделни символи в низа.

Функцията strstr() връща указател към първото появяване на str2 в str1.

Ако не намери съвпадение, функцията връща NULL.

Ако дължината на str2 е 0, функцията връща str1.

Когато strstr() намери съвпадение, можете да получите отместването на str2 в str1 чрез изваждане на указател, както е при strchr().

Процедурата за съвпадение, която strstr() използва, е чувствителна към малките букви.

CODE ACADEMY

Използване на strchr() за търсене на един символ в string .

```
#include <stdio.h>
#include <string.h>
int main()
{
 char *loc, buf[80];
 int ch;
 printf("Enter the string to be searched: ");
 gets(buf);
 printf("Enter the character to search for: ");
 ch = getchar();

 loc = strchr(buf, ch);
 if ( loc == NULL )
   printf("The character %c was not found.", ch);
 else
   printf("The character %c was found at position %d.\n", ch, loc-buf);
 //loc-buf → от адреса на loc вади адреса на buf и получава
 //          относителното отместване = позицията на символа в стринга
return(0);
}
```

CODE ACADEMY

Търсене на набор от символи с strcspn().

```
#include <stdio.h>
#include <string.h>
int main()
{
  char buf1[80], buf2[80];
  size_t  loc;

  printf("Enter the string to be searched: ");
  gets(buf1);
  printf("Enter the string containing target characters: ");
  gets(buf2);

  loc = strcspn(buf1, buf2);
  if ( loc == strlen(buf1) )
       printf("No match was found.");
  else
     printf("The first match was found at position %d.\n", loc);
return(0);
}
```

CODE ACADEMY

Използване на strstr() за търсене на един низ в друг.

```
#include <stdio.h>
#include <string.h>
main()
{
char *loc, buf1[80], buf2[80];

printf("Enter the string to be searched: ");
gets(buf1);
printf("Enter the target string: ");
gets(buf2);

loc = strstr(buf1, buf2);
if ( loc == NULL )
    printf("No match was found.\n");
 else
   printf("%s was found at position %d.\n", buf2, loc-buf1);
return(0);
}
```

Вградени функции от string.h

`char *strrev(char *str);` обръща реда на всички символи в string

`int atoi(char *ptr);`

Конкатенира string **`ct`** към края на string **`s`**, връща като резултат **`s`**,преобразува низа, посочен от ptr, в цяло число.

Освен цифри, низът може да съдържа водещо празно пространство и знак + или -.

Преобразуването започва в началото, преобразува низа, посочен от ptr, в цяло число.

long atol(char *ptr);

double atof(char *str);

Заглавният файл CTYPE.H

CODE ACADEMY

съдържа прототипите за редица функции, които тестват символи, връщайки TRUE или FALSE в зависимост от това дали символът отговаря на определено условие.

isalnum()	Returns TRUE if ch is a letter or a digit.
isalpha()	Returns TRUE if ch is a letter.
isascii()	Returns TRUE if ch is a standard ASCII character (between 0 and 127).
iscntrl()	Returns TRUE if ch is a control character.
isdigit()	Returns TRUE if ch is a digit.
isgraph()	Returns TRUE if ch is a printing character (other than a space).
islower()	Returns TRUE if ch is a lowercase letter.
isprint()	Returns TRUE if ch is a printing character (including a space).
ispunct()	Returns TRUE if ch is a punctuation character.
isspace()	Returns TRUE if ch is a whitespace character (space, tab, vertical tab, line feed, form feed, or carriage return).
isupper()	Returns TRUE if ch is an uppercase letter.
isxdigit()	Returns TRUE if ch is a hexadecimal digit (0 through 9, a through f, A through F).

CODE ACADEMY

Character functions in *C*

<ctype.h>

These return or *false* (0) or *true* (non-zero)

```
int isdigit(int c)   int isalpha(int c)
int isalnum(int c)   int isxdigit(int c)
int islower(int c)   int isupper(int c)
int isspace(int c)   int iscntrl(int c)
int ispunct(int c)   int isprint(int c)
int isgraph(int c)
```

These change case (if appropriate)

```
int toupper(int c)   int tolower(int c)
```

String Conversion Functions in *C*

```
<stdlib.h>

double atof(const char *s)
int atoi(const char *s)
long atol(const char *s)

double strtod(const char *s, char **endp)
long strtol(const char *s, char **endp, int base)
unsigned long strtoul(const char *s, char **endp, int base)
```

CODE ACADEMY

Typical usage of `malloc()` and `free()`

```
char *getTextFromSomewhere(…);

int main(){
  char * txt;
  …;
  txt = getTextFromSomewhere(…);
  …;
  printf("The text returned is %s.", txt);
  return 0;
  free(txt);
}
```

CODE ACADEMY

Typical usage of `malloc()` and `free()`

```
char * getTextFromSomewhere(…){
   char *t;
   ...
   t = malloc(stringLength);
   ...
   return t;
}

int main(){
   char * txt;
   …;
   txt = getTextFromSomewhere(…);
   …;
   printf("The text returned is %s.", txt);
   free(txt);
}
```

CODE ACADEMY

Манипулиране на String

Почти всички C програми, които манипулират текст, правят това с неправилно локирана и освободена памет

Няма ограничение за размера на низа в C

Трябва да сте наясно с размерите на символните масиви!

Трябва да запомните да освободите място за съхранение, когато вече не е необходимо

Преди да забравите показалеца към това хранилище!

An Example of Manipulating String with scanf and printf

```
1.  #include <stdio.h>
2.
3.  #define STRING_LEN   10
4.
5.  int
6.  main(void)
7.  {
8.          char dept[STRING_LEN];
9.          int  course_num;
10.         char days[STRING_LEN];
11.         int  time;
12.
13.         printf("Enter department code, course number, days and ");
14.         printf("time like this:\n> COSC 2060 MWF 1410\n> ");
15.         scanf("%s%d%s%d", dept, &course_num, days, &time);
16.         printf("%s %d meets %s at %d\n", dept, course_num, days, time);
17.
18.         return (0);
19. }

Enter department code, course number, days and time like this:
> COSC 2060 MWF 1410
> MATH 1270 TR 800
MATH 1270 meets TR at 800
```

The `dept` is the initial memory address of the string argument. Thus we don't apply the `&` operator on it.

Some String Functions from String.h

Function	Purpose	Example
`strcpy`	Makes a copy of a string	strcpy(s1, “Hi”);
`strcat`	Appends a string to the end of another string	strcat(s1, “more”);
`strcmp`	Compare two strings alphabetically	strcmp(s1, “Hu”);
`strlen`	Returns the number of characters in a string	strlen(“Hi”) returns 2.
`strtok`	Breaks a string into tokens by delimiters.	strtok(“Hi, Chao”, “ ,”);

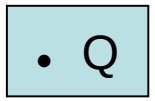
- **Functions `strcat` and `strlen`**

  - **Functions `strcat` and `strncat` concatenate the fist string argument with the second string argument.**
    - `strcat(dest, "more..");`
    - `strncat(dest, "more..", 3);`
  - **Function `strlen` is often used to check the length of a string (i.e., the number of characters before the fist null character).**
    - **e.g., dest[6] = "Hello";**
      strncat(dest, "more", 5-strlen(dest));
      dest[5] = '\0';

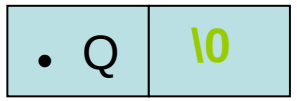
Разлика между символи и стрингове

Представянето на char (т.е., ‘Q’) и на string (т.е., “Q”) е съществено различно.

- **String – това е масив от символи, завършващ с null.**



- Character ‘Q’



- String “Q”

Сравняване на стрингове

- Нека имаме два стринга **`str1`** и **`str2`**.
  - Сравнението **`str1 < str2`** сравнява **initial memory address** на **`str1`** и на **`str2`**.
- Сравнението на два стринга се прави като се сравнява всеки от символите в тях, които си съответстват.
  - Символите се сравняват отново с ASCII таблицата.
  - **“thrill” < “throw” since ‘i’ < ‘o’;**
  - **“joy” < joyous“;**
- Стандартното сравняване на стрингове използва функциите **`strcmp`** и **`strncmp`**

- String Comparison (2/2)

Relationship	Returned Value	Example
`str1 < str2`	Negative	“Hello”< “Hi”
`str1 = str2`	0	“Hi” = “Hi”
`str1 > str2`	Positive	“Hi” > “Hello”

-
```
if(strcmp(str1, str2) != 0)
  printf(“The two strings are different!”);
```

Character Analysis and Conversion

- **<ctype.h>**

Functions	Description
`isalpha`	Check if the argument is a letter
`isdigit`	Check if the argument is one of the ten digits
`isspace`	Check if argument is a space, newline or tab.
`tolower`	Converts the lowercase letters in the argument to upper case letters.

Конвертиране между String и числа

The <stdlib.h> defines some basic functions for conversion from strings to numbers:

- **atoi(“123”) converts a string to an integer.**
- **atol(“123”) converts a string to a long integer.**
- **atof(“12.3”) converts a string to a float.**

- **Няма функции, правещи обратното като itoa, itof, …,**

CODE ACADEMY

Задачи

1. Напишете програма, която въвежда и принтира елементите на масив, използвайки указател.
2. Напишете програма, която копира един масив в друг, използвайки указатели.
3. Напишете програма, която разменя елементите на два еднакви по размер масива, използвайки указатели.
4. Напишете програма, която обръща местата на елементите в един масив, използвайки указатели.
5. Напишете програма за търсене на определен елемент в даден масив, използвайки указатели.

CODE ACADEMY

Задачи

6. Напишете програма, която намира дължината на въведен от клавиатурата стринг, използвайки указатели.
7. Напишете програма, която конкатенира два стринга, използвайки указатели.
8. Напишете програма, която събира две числа, използвайки указатели.
9. Напишете програма, която разменя две числа, използвай указатели.
10. Напишете програма, която обръща стринг, въведен от клавиатурата, използвайки указатели.
11. Напишете програма, която сортира масив, въведен от клавиатурата, използвайки указатели.
12. Напишете примерна програма, която връща няколко стойности от функция, използвайки указател.

CODE ACADEMY

Задачи

Напишете програма, която в даден стринг заменя всички малки букви с главни, всички главни букви с малки, а останалите символи не ги променя.

Програма, която обръща реда на думите в един стринг

Пример, “vali dugd v pet chasa”

“chasa pet v dugd vali”

CODE ACADEMY

Използване на strchr() за търсене на един символ в string .

```
 #include <stdio.h>
#include <string.h>
#define MAX_SIZE 100

int main()
{
    char str[100], reverse[100];
    int len, i, index, wordStart, wordEnd;

    printf("Enter any string: ");
    gets(str);

    len   = strlen(str);
    index = 0;

    wordStart = len - 1;
    wordEnd   = len - 1;
```

```
while(wordStart > 0)
    {
        // If a word is found
        if(str[wordStart] == ' ')
        {
 // Add the word to the reverse string
            i = wordStart + 1;
            while(i <= wordEnd)
            {
                reverse[index] = str[i];

                i++;
                index++;
            }
            reverse[index++] = ' ';

            wordEnd = wordStart - 1;

               }

        wordStart--;
    }// Finally add the last word
    for(i=0; i<=wordEnd; i++)
    {
        reverse[index] = str[i];
        index++;
    }
```

```
// Add NULL character at the end of reverse string

    reverse[index] = '\0';

printf("Original string \n%s\n\n", str);

printf("Reverse ordered words \n%s",
reverse);

    return 0;
}
```

CODE ACADEMY

Задачи

Напишете програма, която намира колко пъти в един стринг се среща символа точка ‘.’
strchr(s,c)

В даден стринг да се намери последното срещане на даден символ
Strrchr()

Програма, която проверява дали първите n символа в един стринг съвпадат с последните му n символа. Числото n се въвежда

strncmp(s1,s2,n) Сравнява първите n символа
strncpy( )

CODE ACADEMY

Задачи

Напишете програма, която в даден стринг
намира определен подстринг и навсякъде, където се среща този подстринг го замества с стринга “abc”

Пример вход стринг “asdasdefwfvasjkwdj”
Търсите подстринг “as” и навсякъде, където го намерите го заместват с “abc”

“abcdabcdefwfvabcjkwdj”

CODE ACADEMY

C – String functions

strlen - Finds out the length of a string
strlwr - It converts a string to lowercase
strupr - It converts a string to uppercase
strcat - It appends one string at the end of another
strncat - It appends first n characters of a string at the end of another.
strcpy - Use it for Copying a string into another
strncpy - It copies first n characters of one string into another
strcmp - It compares two strings
strncmp - It compares first n characters of two strings
strcmpi - It compares two strings without regard to case ("i" denotes that this function ignores case)
stricmp - It compares two strings without regard to case (identical to strcmpi)
strnicmp - It compares first n characters of two strings, Its not case sensitive
strdup - Used for Duplicating a string
strchr - Finds out first occurrence of a given character in a string
strrchr - Finds out last occurrence of a given character in a string
strstr - Finds first occurrence of a given string in another string
strset - It sets all characters of string to a given character
strnset - It sets first n characters of a string to a given character
strrev - It Reverses a string

CODE ACADEMY

Задачи за домашна работа - 1

1) Напишете С програма за намиране на дължината на низ с помощта на цикъл, без да използвате вградена библиотечна функция strlen() .

2) Напишете С програма за свързване на два низа в един низ, без да използвате библиотечна функция strcat().

3) Напишете С програма за сравняване на два низа, използвайки цикъл символ по символ, без да използвате вградена библиотечна функция strcmp()

4) Напишете програма на С, за да проверите дали даден низ е палиндром или не, без да използвате цикъл.

5) Напишете С програма за намиране на символа с най-висока честота в низ с помощта на цикъл.

CODE ACADEMY

Задачи за домашна работа - 2

6) Напишете С програма за премахване на всички повтарящи се символи в низ с помощта на цикли.

7) Напишете С програма за отрязване както на водещите, така и на крайните символи за празно пространство в низ с помощта на цикъл.

1) Напишете С програма за премахване на допълнителните интервали и празни места от низ.

2) Напишете С програма за броене на срещания на дума в даден низ с помощта на цикъл.

3) Напишете С програма за премахване на всички срещания на дадена дума в низ с помощта на цикъл.