

**Тема: Работа с `printf()` и `scanf()`.**

### **1. Език за програмиране C.**

C е език за програмиране от средно ниво. Това означава, че има управляващи конструкции и изрази, характерни за езиците от високо ниво, но в същото време осигурява механизми за достъп до паметта, подобно на машинните и асемблерни езици.

Програмите са написани в четим от човека код, който не може да се изпълни директно от компютър. Текстът на програмата (сорс кода, source code) се съдържа в сорс файлове, в C са с разширение .c. Типът на файла е важен, за да го разпознае и обработи компилатора. Функцията `main()` е задължителен компонент - мястото, от което започва всяка C програма.

Необходим е процес от три стъпки за трансформиране на сорс кода в програма, обработваща данни. Този процес е характерен за почти всички съвременни езици. В повечето съвременни среди за програмиране тези три стъпки се обработват от едно приложение. Трите стъпки са: компилиране, свързване и изпълнение.

- **Компилиране (*compiling*)** - Компилирането включва предварителна стъпка – обработка от препроцесора. Препроцесорът обработва директиви (команди, които започват със знак #), които могат променят сорс кода, преди да бъде компилиран. Сорс кодът се компилира до двоичен обектен код (файл с разширение .obj). Този код все още не е изпълним.
- **Свързване (*linking*)** - На този етап обектният код може да се свърже с друга обектна програма, за да се получи изпълнима програма. На този етап се включват библиотеките. Резултатът от свързването е изпълним файл - exe.
- **Изпълнение (*running*)** - На този етап готовото вече приложение обработва зададените входни данни, което би могло да включва и интерактивно взаимодействие с потребителя.

## 2. *printf()* и *scanf()*.

Стандартът ANSI C определя набор от функции, наречени стандартна библиотека, които са част от всеки C компилатор.

Едни от най-използваните функции на стандартната библиотека са *printf()* и *scanf()*. За да ги извикаме необходимо да включим хедърния файл `stdio.h` в нашата програма.

Библиотечната функция *printf()* се използва за извеждане в конзолата. Тя може да приеме единичен низ от символи както в случая

*printf("Hello world")*

Или дефиниран от програмиста брой аргументи;

*printf("Контролен низ", ..., .....,...).*

В случая когато програмистът дефинира броя на аргументите, първият аргумент винаги е контролен низ, съдържащ форматни спецификатори. Всички форматни спецификатори започват с %. Форматният спецификатор, определя как ще бъде изведен аргумента, асоцииран с него. Форматните спецификатори и техните аргументи се асоциират отляво надясно и е много важно да има толкова аргументи, колкото са спецификаторите. Форматните спецификатори са обвързани с типа данни.

Контролния низ може да съдържа също текст, както и няколко специални знакови кода (за авички, наклонена черта, звуков сигнал).

Най-често използваната функция за въвеждане на стойности през конзолата е *scanf()*. Първият аргумент винаги е контролен низ, който определя как да се обработват следващите аргументи. Контролният низ се състои от форматни спецификатори, те са същите, както при *printf()*. Следващите аргументи са адреси на променливи, които ще съхраняват въведените от потребителя стойности. Адрес на променлива в C се задава чрез *&име\_на\_променлива*.

Таблицата съдържа няколко основни типа данни и техните форматни спецификатори.

Data Type	Size (bytes)	Format Specifier
<b>int</b>	4	%d
<b>char</b>	1	%c
<b>float</b>	4	%f
<b>наз</b>	променлив	%s

### 3. Променливи.

Променливата е именувана част от паметта, която съхранява стойност. Стойността може да бъде променяна в хода на изпълнение на програмата.

Имената на променливите в С се състоят от букви от латинската азбука, цифрите от 0 до 9 и `_`, като името не може да започва с цифра. Езикът С е чувствителен към регистъра на буквите. *A* и *a* са две съвсем различни имена на променливи.

Декларацията на променлива задава типа на променливата, тоест какви стойности може да съхраняваме в нея. Синтаксът е:

***тип име\_на\_променлива;***

Променливите от тип *int* може да съхраняват положителни и отрицателни цели числа (без десетична запетая). Променливите от тип *float* може да съхраняват положителни и отрицателни числа с десетична запетая. Променливите от тип *char* обикновено се използва за съхранение на символ. Тъй като езикът С е много гъвкав, бихме могли да използваме и за съхранение на число.

За да се присвои стойност на променлива е необходимо името ѝ да е от лявата страна на знака за равенство, а стойността – от дясната, например:

***int a=100;***

***char c='V';***

***float f=1.6;***