Съставяне, въвеждане и настройка на програми, илюстриращи операциите в езика и основните типове данни

Структура на програма

<библиотеки>

…деклариране на глобални променливи и спомагателни функции

<главна функция>

{

< тяло на функцията>

}

**<библиотеки>** - в тях се съдържат основни и спомагателни методи, оператори и функции по подразбиране. Библиотеките се изброяват една под друга, като започват **с #include** след което се записва пълен път до файла, който искаме да използваме за библиотека, а ако е от подразбиращите се стандартизирани в езика само името му. Обикновено библиотеките завършват с **.h**, което идва от така наречените **header** файлове – спомагателни с готов за ползване код. Header файлове може да направите и сами, като за използването им е нужно да спазите гореописаните правила.

Когато е от стандартните библиотеки то извикване по следния начин:

**#include <име на библиотека .h>**

Когато е създадена от нас:

**#include “пълен път и име на файла, който ще използваме”**

Между библиотеката и главната функция могат да бъдат декларирани и описвани други функции, които играят спомагателна роля за работата на програмата.

Променливите описани извън функции в основното тяло на програмата се наричат ***глобални***. За тях ще стане дума в следващата точка от упражнението.

**<главна функция>** - главната функция – **main** – е основната функция, от която започва работата на програмата. Тя е задължителен елемент от програмата и може да съществува само една. Всяка друга функция играе роля на спомагателна към основната и развива изграждането на програмния код.

**<тяло>** на функция или оператор е мястото, което се огражда с **{ }**, между които пишем код с определена функционалност. Цялото съдържание на кода между скобите се изпълнява при едно логическо използване на оператора или функцията, към които е прикачен. Възможно е да имаме множество вграждания на <тяло> в <тяло>, като се спазва правилото, че скобата за затваряне - } – затваря последното отворено място – {. За да функционира правилно програмата трябва всички тела да са правилно отворени и затворени.

Пример за правилно написана програма:

**#include <stdio.h>**

**int main()**

**{**

**printf(“Hello World!\n”);**

**return 0;**

**}**

Понятие за променлива

Променливата в програмните езици има за цел да съхранява данни. За да се използва променлива в програмният език С, са необходими две неща: да се избере подходящ тип на данните – това е видът данни, който ще се съхранява в тази променлива, както и име на самата променлива. Записването на този избор в програмния код се нарича дефиниция. Всяка променлива, използвана в С, задължително трябва да бъде дефинирана. По този начин, компилаторът ще знае колко памет да задели и как да интерпретира тези данни.

Когато на променлива даваме стойност то тогава декларираме самата променлива като тя приема в паметта съответната стойност.

**Видимост на променлива** – в езика С видимост на променлива или живот означава мястото, на което може да използваме вече декларирана или дефиниране променлива.

***Тук се влияем от няколко правила:***

-Всяка променлива е видима след мястото си на дефиниране/деклариране, съответно може да бъде ползвана от там нататък в кода.

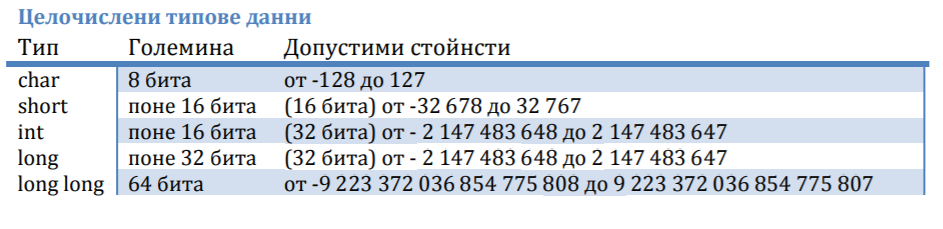
-Глобалните променливи се дефинират извън функциите в общата част на програмата, имат стойност 0 при дефиниция и могат да бъдат ползвани навсякъде в програмата след дефиницията им.

-Локалните променливи се използват само в блока код, в който са обявени и нямат стойност по подразбиране по време на дефиниция. Техният живот е в рамките на обявеното тяло. Локалните променливи са с по-голям приоритет при ползване в кода от глобалните.

*Не могат да съществуват променливи с едни и същи имена и от един и същи вид в едно тяло!*

Видове променливи в С

В програмния език С са създадени типове данни, които служат за изграждането на основата за съхранението на данните. Основните типове данни имат за цел да описват цели числа, реални числа и букви. Всеки един тип от тези данни се характеризира със следните параметри – размер байтове, които заема в паметта; интервал на стойностите, които могат да бъдат записвани чрез този тип.



В таблицата са показани допустимите стойности за всеки един тип за компилатор GCC, разликите се в допустимите стойности се наблюдават, защото стандартът за езика С е гъвкав и има възможност да варират стойностите за различните компилатори.

Картина, която съдържа текст

Описанието е генерирано автоматично

От таблицата се вижда, че числата в плаваща запетая като допустими стойности се различават драстично. Освен това типове като double и long double се препоръчва да се използват, когато точността на реалните числа е от значение.

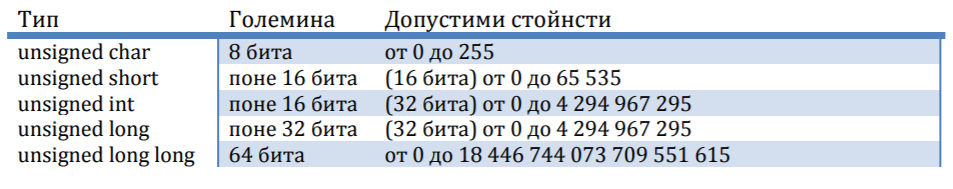
*Типовете данни, които описват реалните числа се наричат „числа с плаваща запетая“, защото в представянето на числата в компютъра се извършва използвайки записване чрез мантиса и порядък, които се записват на различни места в двоичния запис на числото*

Типове от данни описващи букви

В програмният език С съществува един единствен тип данни, който описва букви и символи. Този тип е char (съкратено от character от англ. „буква, знак“). Този тип както беше споменато е и тип, описващ целочислени числа. Причината типът да има така дуалност (двойнствен характер) се крие в представянето на символите в една компютърна система. Всички възможно символи са записани в една таблица (ASCII таблица), в която срещу всеки символ е записано съответното му число. Така, когато трябва стойността на променливата от тип char да бъде представена като буква, компютърът автоматично заменя числовата ѝ стойност, със символа записан срещу нея. По този начин символите като ‘A’ имат стойност 65, а ако към буквата прибавим число получим символ, който седи на отстояние толкова места, колкото е стойността на числото.

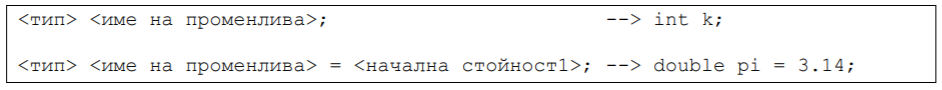
Безнакови типове данни

В програмният език С има типове, които описват само цели положителни числа. Тези типове данни се получават като пред целочислените типове данни се запише unsigned. По този начин новите типове имат следните допустими стойности:



Създаване и основни операции с променливи

Създаване на променливи Дефинирането на променливи се извършва в началото на тялото на всяка функция, според стандарта на С. За да бъде създадена една променлива е нужно да се следва един от следните два модела:



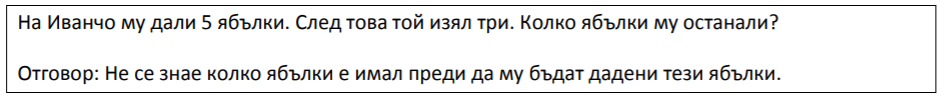
В двата модела се заместват полетата както следва:

**<тип>** - това поле се замества с желаният тип, който е нужен за съхраняване на данните;

**<име на променлива>** - това е името на данните, което ще се използва за достъп то тези данни;

**<начална стойност>** - може да се зададе начална стойност на дадената променлива.

Важно е да се отбележи, че в програмният език С компилаторът не се грижи за зануляване на стойностите при тяхното дефиниране, за това е добра практика да се задава начална стойност. Този проблем може да се демонстрира и със следната логическа задача:



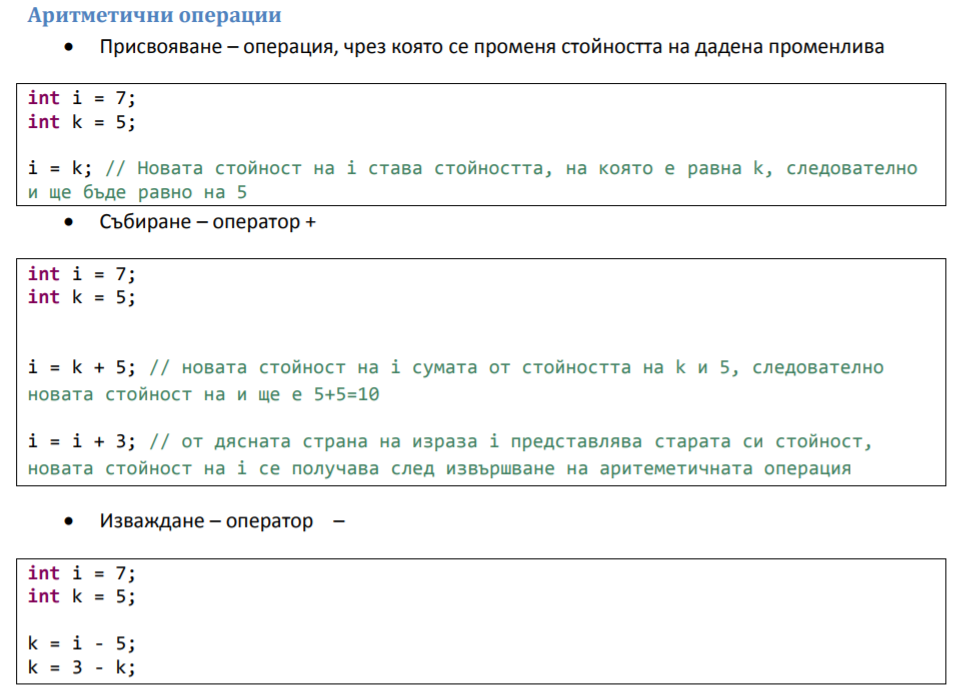
На един ред може да има дефинирана повече от една променлива от един и същи тип. Имената на променливите се разделят със запетая.

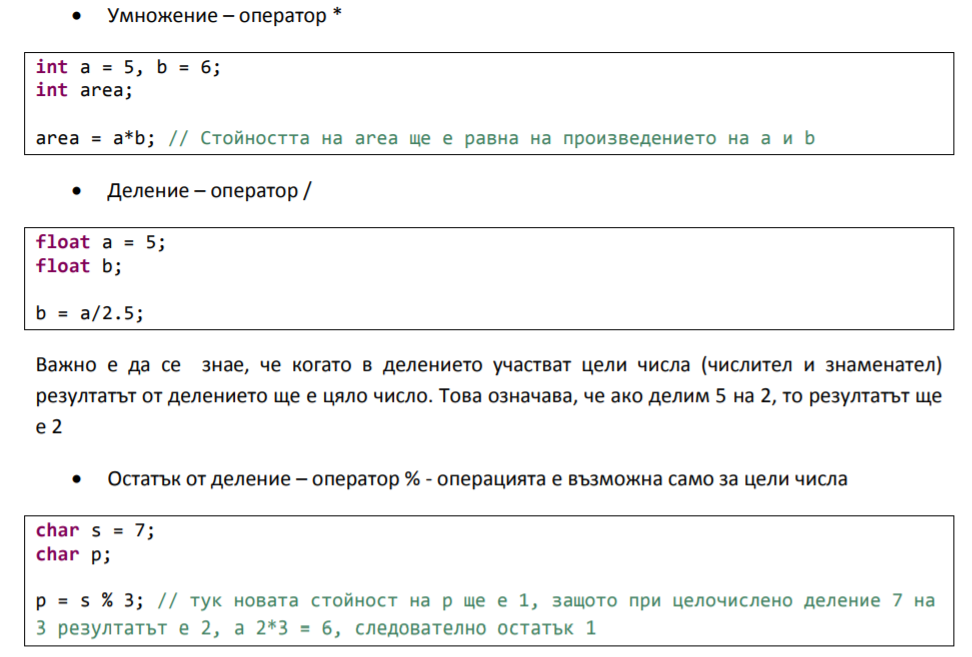
***Задача: Съставете програма на С, като използвате дефиниция и декларация. Пробвайте различните типове данни.***

***Задача: Изведете цифрите от ASCII таблицата, направете същото и за малки и големи букви по избор.***

Операции с елементарен тип данни

Операциите с елементарните типове данни се разделят на няколко типа - аритметични, логически и побитови (последният вид само за целочислени променливи).





***Задачи:***

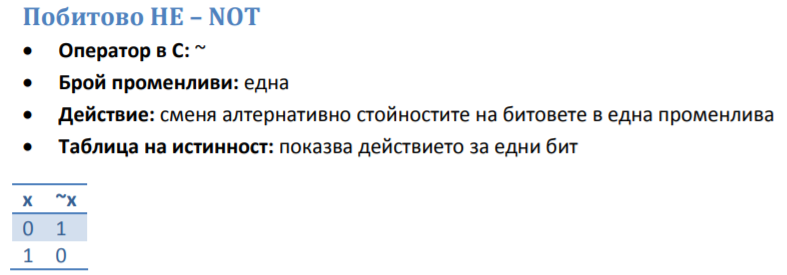
***1. Да се напише програма, която намира лице на правоъгълник по зададени от клавиатурата две страни и го извежда на екрана.***

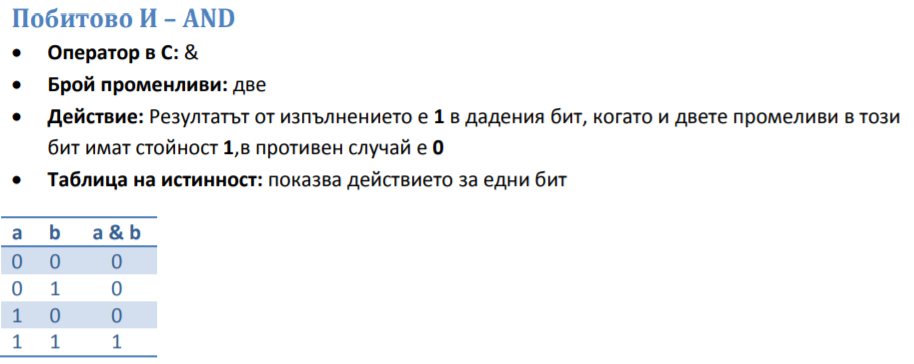
***2. Да се напише програма, която намира обиколката на окръжност по зададен диаметър от клавиатурата и извежда резултата на екрана.***

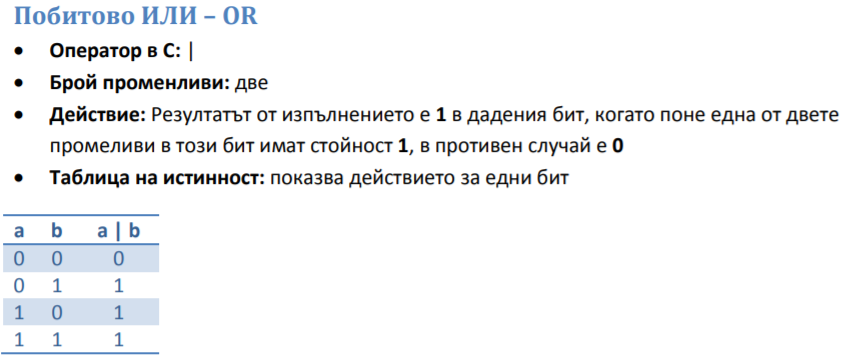
Побитови операции в С

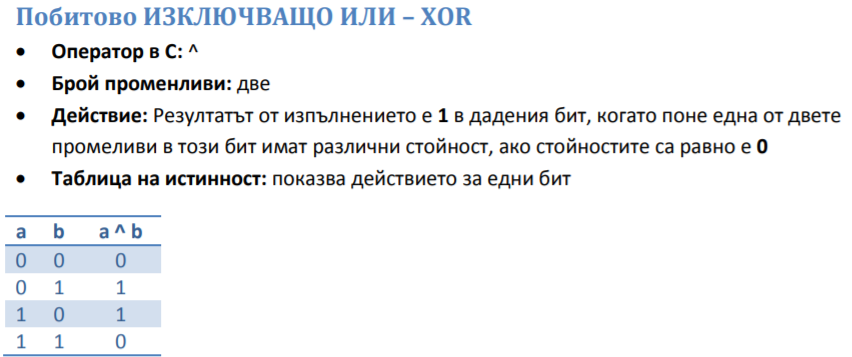
Програмният език С позволява работа с оператори за побитови операции: **NOT, AND, OR, XOR, SHIFT LEFT, SHIFT RIGHT**

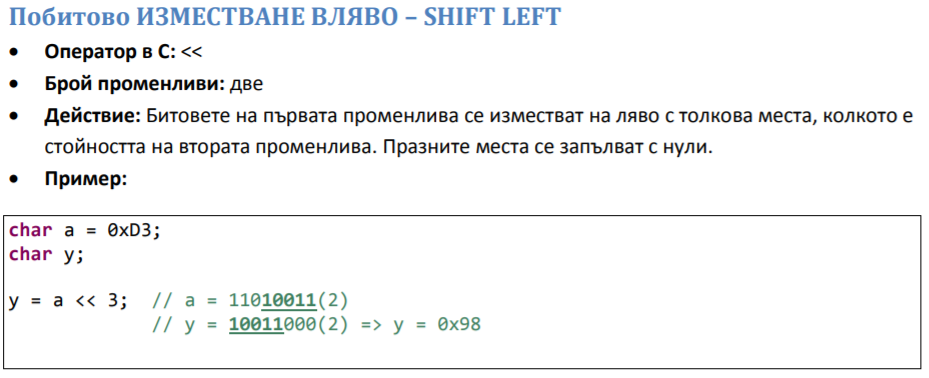
Действията се извършват на ниво битове в двоичния запис на числото.

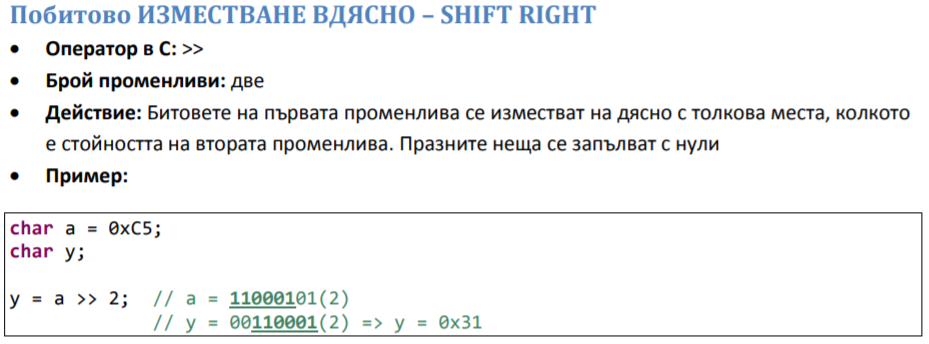












Логически оператори в езика С

Логическите операции са част от операциите, с които може да се работи с елементарните типове данни.

В програмният език С под „истина“ се разбира ненулева стойност, а под „лъжа“ – нула.

