**Л Е К Ц И Я 5**

**Ключови думи: if, else, case, default**

**Условни оператори в езика С**

**IF**

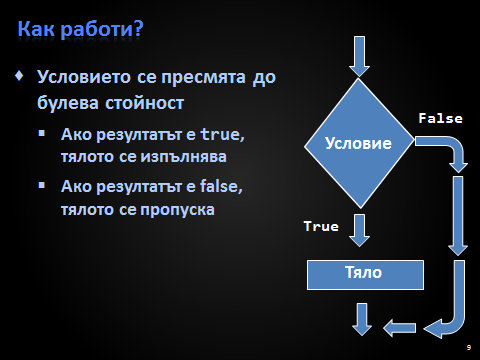
В компютърните науки, **условните конструкции** са функции на [езика за програмиране](https://bg.wikipedia.org/wiki/%D0%95%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5), чрез които можем да изпълняваме различни действия в зависимост от някакво условие. В езиците от семейството на C разграничаваме следните видове условни конструкции: **if / else / else if / вложени if** конструкции. Една програма може да се държи различно, в зависимост от някакво условие посредством условните конструкции if и if-else. Те представляват тип условен контрол, който се проверява по време на изпълнение на конструкцията. Форматът на условната конструкция if е следният: if-клауза, булев израз и тяло на условната конструкция:

**if** (булев израз)

{

тяло на условната конструкция;

}



Ако изразът в скобите след ключовата дума **if** бъде изчислен до стойност true, се изпълнява тялото на условната конструкция. Ако резултатът от изчислението на булевия израз е false, то операторите в тялото няма да бъдат изпълнени. Пример:

**if (n > 0) {**

**if (a > b)**

**z = a;**

**}**

**else**

**z = b;**

Условие

**Тяло**

**Тяло**

Условният оператор **if** се използва за сравняване на стойностите на променливи и извършване на действия спрямо резултата. Например може да сравните дали стойността въведена от потребителя е равна на дадена стойност:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int age;**

**cout << "Enter your age: ";**

**cin >> age;**

**if (age == 18)**

**cout << "You are 18 years old";**

**return 0;**

**}**

В горния пример виждате, че за сравняване на две стойности се използва ==, а не = защото едното равно се използва за присвояване на стойности в променливи. Ако потребителя въведе числото 18, тогава на екрана ще се отпечата "You are 18 years old". Това са всички оператори който могат да ви послужат за сравняване на 2 стойности:

== Равно

!= Не е равно ( ! пред = се използва за отрицание )

> По-голямо

>= По-голямо или равно

< По-малко

<= По-малко или равно

Ако искате да напишете повече от една команда за изпълнение в IF, трябва да използувате фигурни скоби ето така:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int age;**

**cout << "Enter your age: ";**

**cin >> age;**

**if (age == 18)**

**{**

**cout << "You are ";**

**cout << "18 years old";**

**}**

**return 0;**

**}**

**ELSE**

IF проверява условието и връща резултат "true" или "false". Ако въведените години от потребителя са 18, то тогава ще върне "true", ако не са 18 ще върне "false". Ако искате когато върне "false" пак да се изпълнява някакъв код, то тогава трябва да добавите ELSE, ето така:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int age;**

**cout << "Enter your age: ";**

**cin >> age;**

**if (age >= 18)**

**cout << "You are an adult";**

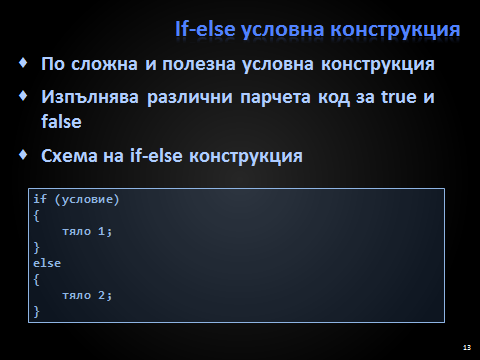
**else**

**cout << "You are a child";**

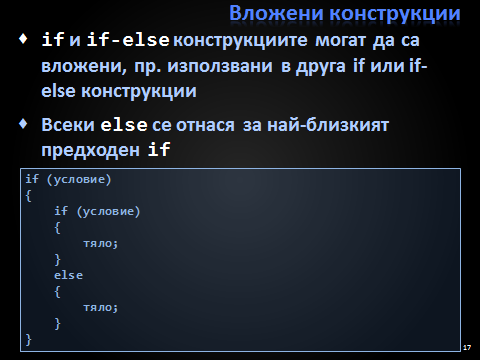
**return 0;**

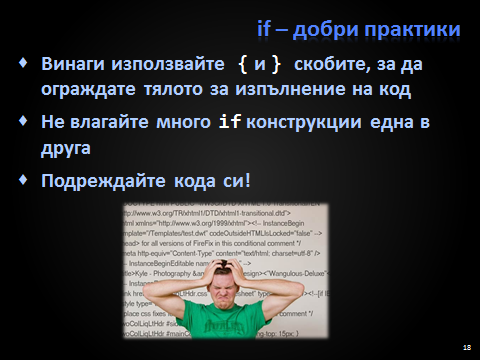
**}**

Сега в този пример виждаме, че ако потребителя въведе число по-голямо или равно на 18, IF ще върне "true" и кода под него ще се изпълни, тоест на екрана ще види "You are an adult", но ако числото е по-малко от 18 тогава ще върне "false" и влиза в сила ELSE, на екрана ще се отпечата "You are a child".









Могат да се проверяват и повече условия наведнъж, това става като използваме &&(и) или ||(или). Например, ако между 2 условия сложите && за да върне "true" трябва и двете да са изпълнени ако едното или и двете не са изпълнени връща "false", докато ако сложите || е нужно само едно от 2-те условия да е изпълнено за да върне "true", в този случай ще върне "false" само ако и двете условия са неизпълнени. Може да тествате сами тази програма:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int num1, num2;**

**cout << "Enter first number: ";**

**cin >> num1;**

**cout << "Enter second number: ";**

**cin >> num2;**

**if (num1 == 1 && num2 == 2)**

**cout << "You entered the right values";**

**if (num1 == 1 || num2 == 2)**

**cout << "One of the values is right";**

**if (!(num1 == 1))**

**cout << "You entered the wrong value";**

**return 0;**

**}**

Ето още един пример:

**#include <stdio.h>**

**/\* print Fahrenheit-Celsius table \*/**

**main()**

**{**

**int fahr;**

**for (fahr = 0; fahr <= 300; fahr = fahr + 20)**

**printf("%3d %6.1f\n", fahr, (5.0/9.0)\*(fahr-32));**

**}**

**Тернарен оператор ? :**

Условният [оператор](https://bg.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)) ? : е оператор в езика [C](https://bg.wikipedia.org/wiki/C_(%D0%B5%D0%B7%D0%B8%D0%BA_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)) и C-подобните езици. Известен е още като тернарен оператор, тъй като е единствения оператор, който приема 3 [операнда](https://bg.wikipedia.org/w/index.php?title=%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D0%BD%D0%B4_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)&action=edit&redlink=1):

**операнд1 ? операнд2 : операнд3**

Първия операнд или условието на условната конструкция може да бъде булева променлива или [булев израз](https://bg.wikipedia.org/w/index.php?title=%D0%91%D1%83%D0%BB%D0%B5%D0%B2_%D0%B8%D0%B7%D1%80%D0%B0%D0%B7&action=edit&redlink=1) и може да приема двете [булеви стойности](https://bg.wikipedia.org/wiki/%D0%9B%D0%B8%D1%82%D0%B5%D1%80%D0%B0%D0%BB#.D0.91.D1.83.D0.BB.D0.B5.D0.B2.D0.B8) true (истина) и false (лъжа). Ако след извършването на необходимите преобразувания, операнд1 се сведе до истинно твърдение (true), то тогава след изпълнението си тернарния оператор ще върне стойността на операнд2, в противен случай (false), върнатата стойност ще бъде стойността на операнд3.

**int a = 5;**

**int b = 3;**

**int larger = (a > b) ? a : b;**

В горния пример, инициализираме 2 [целочислени променливи](https://bg.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8_%D1%82%D0%B8%D0%BF%D0%BE%D0%B2%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8) а и b и им задаваме стойности съответно 5 и 3. На [променливата](https://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BC%D0%B5%D0%BD%D0%BB%D0%B8%D0%B2%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5)) larger присвояваме резултата от тернарния оператор. В случая това е 5, тъй като условието (a > b) е изпълнено (true), върнатата стойност ще бъде тази на операнда преди двете точки, тоест променливата a, която е 5. Този пример показва как по лесен начин можем да определим кое от две числа е по-голямо, без използването на if-else конструкцията. Същия пример, реализиран чрез if-else би изглеждал по следния начин:

**int a = 5;**

**int b = 3;**

**int larger;**

**if (a > b)**

**larger = a;**

**else**

**larger = b;**

**Пример:**

**1.7==5 ? 4 : 3     // връща 3, защото 7 не е равно на 5.**

**2.7==5+2 ? 4 : 3   // връща 4, защото 7 е равно на 5+2.**

**3.5>3 ? a : b      // връща стойността на a, защото 5 е по-голямо от 3.**

**4.a>b ? a : b      // връща по-голямата стойност, или a или b.**

**01.// conditional operator**

**02.**

**03.#include <iostream>**

**04.using namespace std;**

**05.**

**06.int main ()**

**07.{**

**08.  int a,b,c;**

**09.**

**10.  a=2;**

**11.  b=7;**

**12.  c = (a>b) ? a : b;**

**13.**

**14.  cout << c;**

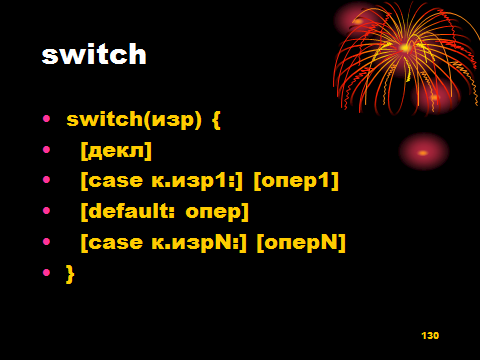
**15.**

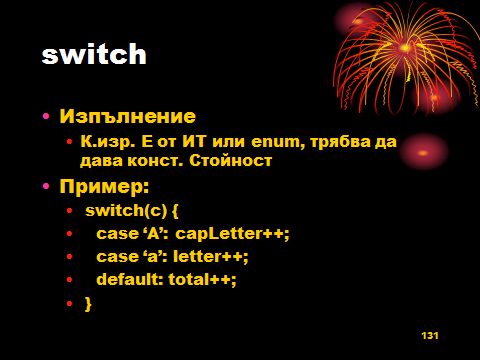
**16.  return 0;**

**17.}**

В този пример *a* беше 2, а *b* 7, затова изразът (a>b) не е верен. Следователно целият израз връща втората стойност (тоест b със стойност 7).

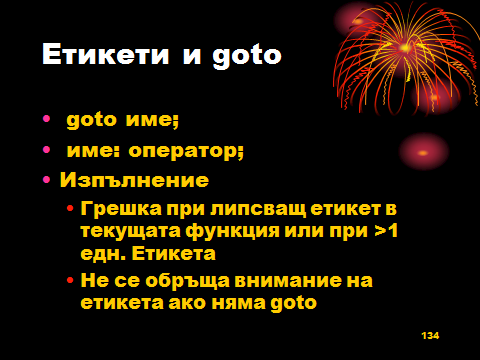
**Оператор Case в Switch**

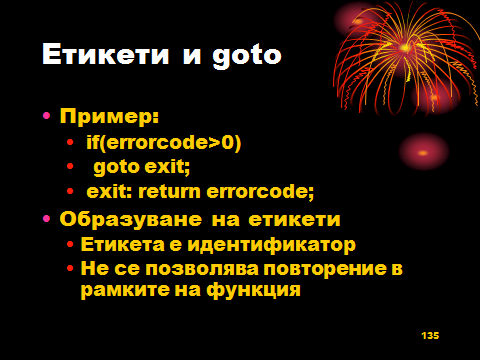
****

****

От примера се вижда, че последния оператор **default** е по подразбиране в случай, че нито едно от условията **case** не е изпълнено.

**Оператор goto**

****

****

**Форматиран вход-изход**

Функциите за форматиран вход-изход printf и scanf са функции с променлив брой аргументи, които не са част от езика С. В него няма дефиниран вход-изход. Това са просто полезни функции от стандартната библиотека на езика. Поведението им е дефинирано в ANSI стандарта.

**int printf (const char \*format [, argument, ...]);**

**int scanf (const char \*format [, address, ...]);**

където:

**const char \*format -** задължителен форматиращ низ; той определя броя на аргументите на двете функции, чрез форматни спецификатори задавани със символа %

общият вид на форматните спецификатори е

**% [флаг] [ширина на полето] [.точност] [модификатор] конвертиращ символ**

|  |  |
| --- | --- |
| Символ за извеждане и въвеждане**:**  **%i или %d** | **int** |
| **%f** | **float** |
| **%c** | **char** |
| **%s** | **Низ от символи**  Някои флагове:  ------------------------------------------------------------------------  **-** подравнява изхода в ляво (по подразбиране е в дясно)  **+** числата със знак се извеждат задължително с + или–  --------------------------------------------------------------------------  **space**  Положителните  числа се извеждат  със space,отрицателни с  със –  ----------------------------------------------------------------------------------------   |  | | --- | | **0** |   при числови стойности  се запълва с 0  **Някои модификатори –** **l за long и double, h за short**  **argument** - списък от аргументи с които ще се извършва обмена**.**  **връщан резултат:**  **print - броя на изведените символи или константата EOF(дефинирана в stdio.h), при неуспешно извеждане;**  **scanf - броя на правилно обработените полета и при прочитане на край на файл - EOF (под правилно обработени полета се разбира, прочитане, преобразуване в указания формат и присвояване на преобразуваната стойност на адреса на програмния обект от списъка за аргументи). Ако не е преобразувано нито едно поле върнатият резултат е 0 (нула).**   |  |  | | --- | --- | | Примери:  **#include<stdio.h>**  **int main(){**  **int a,b,n;**  **float c,d; char z='a';**  **a = 15;**  **b = a / 2;**  **n=printf("%d %3d %03d\n",b,b,b);**  **printf("printed %d char\n",n);**  **n=printf("%+d %-3d\n",b,b);**  **printf("printed %d char\n",n);**  **c = 15.3; d = c / 3;**  **n=printf("%3.2f\n",d);**  **printf("printed %d char\n",n); n=printf("%c %3c %%\n",z,z);**  **printf("printed %d char\n",n);**  **return 0;**  **}**  **Пример:**  **#include<stdio.h>**  **int main(){**  **int a,n; float f; double d;**  **printf("input a,f and d:");**  **n=scanf("%d%f%lf",&a,&f,&d); printf("a=%d f=%5.3f d=%6.4lf n=%d\n",a,f,d,n);**  **return 0;**  **}**  -----------------------------------------------  input a,f and d:45 67.3 86.45  a=45 f=67.300 d=86.4500 n=3  **Внимание: & при scanf - адреси!** | **7 7 007**  **printed 10 char**  **+7 7**  **printed 7 char**  **5.10**  **printed 5 char**  **a a %**  **printed 8 char** | |  |

|  |
| --- |
|  |