

Управляващи структури — оператори за разклонение

conditional / branching statements if-else, switch

Оператор if

if statement

if

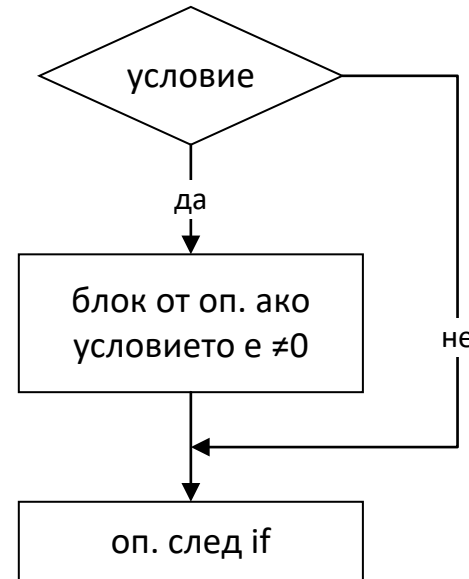
if(условие)

 операция ако усл. е $\neq 0$;
оп. след if;

- След скобата на if **не** се слага точка и запетая ; (освен ако целта не е да не се случи нищо – празна операция)
- Само една операция е подчинена на if оператора
- Блок от операции – между {}, се разглежда като една операция

if(num < 0)

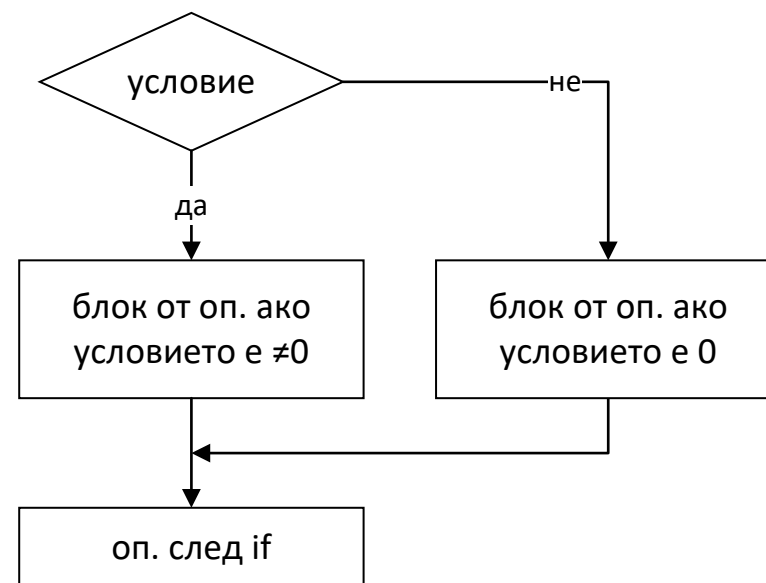
 printf("Value %d is negative.");



if - else

```
if(условие)  
    операция ако усл. е  $\neq 0$ ;  
else  
    операция ако усл. е  $= 0$ ;  
оп. след if;
```

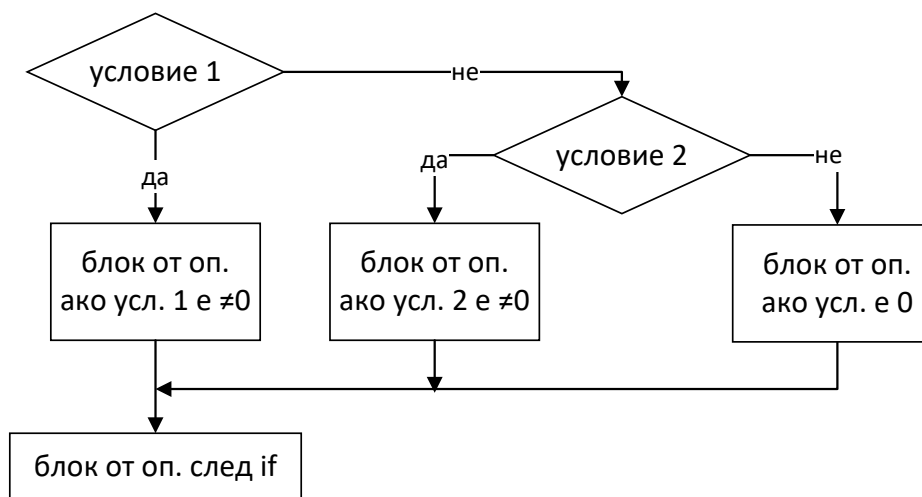
```
if(num % 2)  
{  
    printf("%d е nechetno chislo.");  
}  
else  
{  
    printf("%d е chetno chislo.");  
}
```



Влагане на оператор if в else

```
if(условие 1)
    {блок ако усл. 1 е ≠0;}
else if(условие 2)
    {блок ако усл. 2 е ≠0;}
...
else
    {блок ако всички горни условия
    са =0;}
оп. след if

if(a > b)
    printf(„%d е по-голямо от %d”, a, b);
else if (a < b)
    printf(„%d е по-малко от %d”, a, b);
else
    printf(„dвете числа са равни - %d”, a);
```



Пример if-else-if

```
/* Програма-игра за отгатване на число */
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int number, guess;
    number = rand(); //rand е функция за генериране на псевдо случайно число
    printf("Otgatnete chisloto: ");
    scanf("%d", &guess);
    if (guess == number)
        printf("Poznahte!\n");
    else if (guess > number)
        printf("Greshka, chisloto e mnogo golqmo!\n");
    else
        printf("Greshka, chisloto e mnogo malko!\n");

    printf("Chisloto e %d", number);

    return 0;
}
```

Забележки към примера

- `int rand(void)`; е функция за генериране на псевдо случайни числа (две последователни извиквания на програмата генерира едни и същи числа)
- За да се използва се включва `#include <stdlib.h>`
- Използва се в комбинация с функцията `void srand(unsigned int seed)`, за да генерира различни числа
- За `seed` често се използва времето – заглавен файл `#include <time.h>`, структура `time_t`, функция `time`

Пример с различни случайни числа

```
/* Програма-игра за отгатване на число */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main()
{
    int number, guess;

    srand(time(NULL));
    number = rand() % 100;

    printf("Otgatnete chisloto: ");
    scanf("%d", &guess);
    if (guess == number)
        printf("Poznahte!\n");
    else if (guess > number)
        printf("Greshka, chisloto e mnogo golqmo!\n");
    else
        printf("Greshka, chisloto e mnogo malko!\n");

    printf("Chisloto e %d", number);

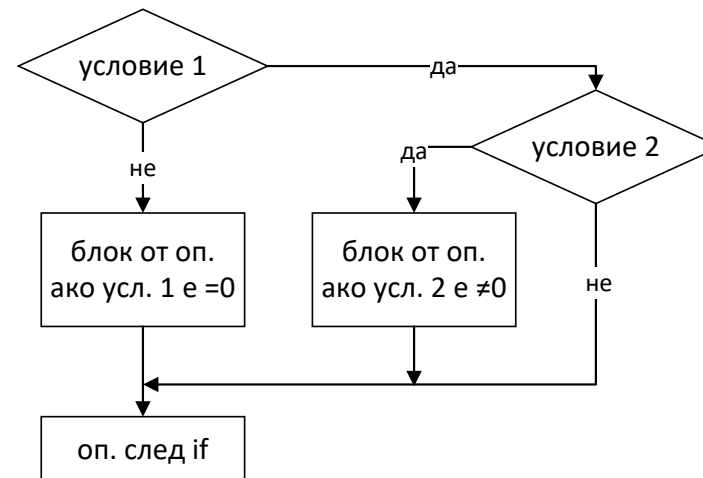
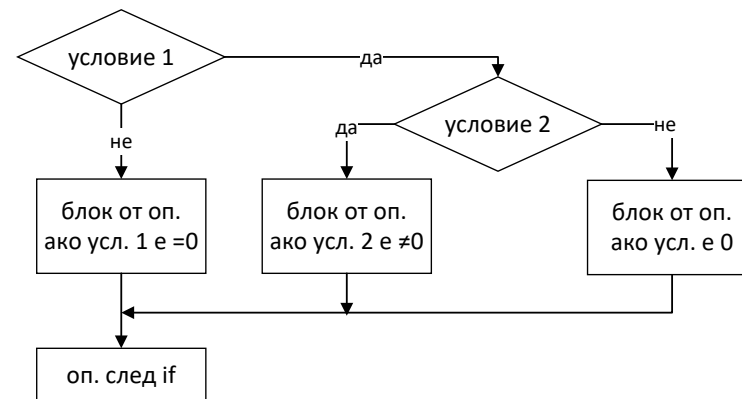
    return 0;
}
```


Влагане на оператор if в if

```
if(num >= 0)
    if(num == 0)
        printf("Stoinostta e nula.");
    else
        printf("Stoinostta e polojitelna.");
else
    printf("Stoinostta otricatelna.");
```

- else се свързва към първия свободен if (за който няма else)

```
if(input)
{
    if(canProcess)
        printf("Processing ...");
}
else
    printf("Waiting for input ...");
```

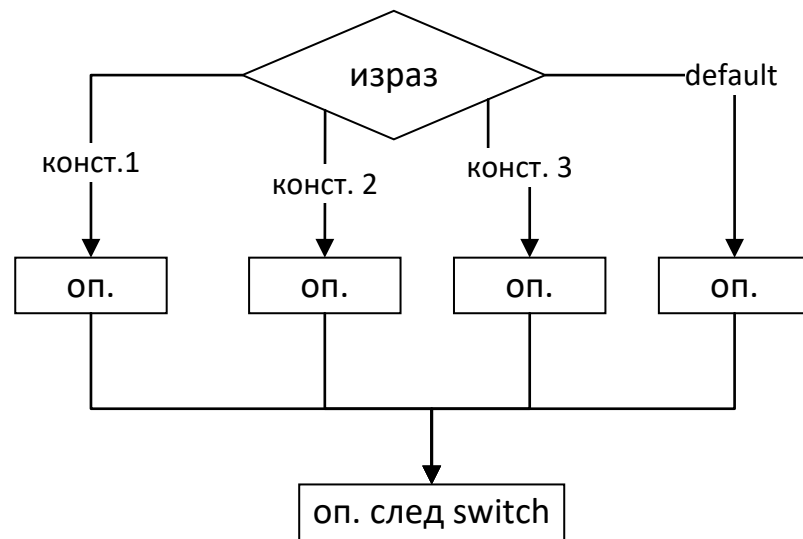


Оператор switch

switch statement

Синтаксис

```
switch(израз)
{
case const1:
    оп1;оп2;...опN;
    break;
case const2:
    оп1;оп2;...опN;
    break;
.
.
default:
    оп1;оп2;...опN;
    break;
}
```



Switch

- Използва се за проверка дали дадена променлива или израз има конкретна стойност
- Проверката става с разглеждане на отделни случаи – case.
- Във всеки case се поставя константна стойност, която се сравнява със стойността на израза
- Във всеки case се описват операциите, които се изпълняват ако изразът има съответната стойност
- Случаят default не е задължителен. Ако го има, се изпълнява ако никой от останалите случаи не отговаря на стойността на израза
- На края на всеки case може да се сложи оператор break;, който указва излизане от switch-a
- Ако няма break на края на case, програмата ще продължи изпълнението със следващия case, докато не стигне до break или до края на switch-a

Програма за определяне на ден от седмицата

```
#include <stdio.h>

int main()
{
    int den;
    printf("Vavedete chislo, poreden den ot sedmicata: ");
    scanf("%d", &den);
    switch (den)
    {
        case 1:
            printf("Denqt e ponedelnik");break;
        case 2:
            printf("Denqt e vtornik");break;
        case 3:
            printf("Denqt e srqda");break;
        case 4:
            printf("Denqt e chetvartak");break;
        case 5:
            printf("Denqt e petak");break;
        case 6:
            printf("Denqt e sabota");break;
        case 7:
            printf("Denqt e nedelq");break;
        default:
            printf("Chisloto ne e den ot sedmicata!");break;
    }
    return 0;
}
```

Пример с пропадане в следващ case (case fall through)

```
switch (month)
{
    case 12:
    case 1:
    case 2:
        printf("Zima"); break;
    case 3:
    case 4:
    case 5:
        printf("Prolet"); break;
    case 6:
    case 7:
    case 8:
        printf("Liato"); break;
    default:
        printf("Esen"); break;
}
```

Програма калкулатор – сравнение if и switch

```
#include <stdio.h>

int main()
{
    float a, b;
    char op;

    printf("Vavedete operandi: ");
    scanf("%f%f", &a, &b);
    printf("Vavedete operator: ");
    op = getchar();

    switch (op)
    {
        case '+':
            printf("Sborat na %.2f i %.2f e %.2f.", a, b, a + b);
            break;
        case '-':
            printf("Razlikata na %.2f i %.2f e %.2f.", a, b, a - b);
            break;
        default:
            printf("%c e nepoddarjana operacij!", op);
            break;
    }
    return 0;
}
```

```
#include <stdio.h>

int main()
{
    float a, b;
    char op;

    printf("Vavedete operandi: ");
    scanf("%f%f", &a, &b);
    printf("Vavedete operator: ");
    op = getchar();

    if(op == '+')
    {
        printf("Sborat na %.2f i %.2f e %.2f.", a, b, a + b);
    }
    else if(op == '-')
    {
        printf("Razlikata na %.2f i %.2f e %.2f.", a, b, a - b);
    }
    else
    {
        printf("%c e nepoddarjana operacij!", op);
    }
    return 0;
}
```

Забележки при четене на символи

- Функциите за четене `scanf` и `getchar` четат от буфера на стандартния вход (`stdin`).
- Символите, с които се задава край на въвеждането на стойност – интервал / символ за нов ред (`enter`), също се записват в този буфер
- При четене на символ в буфера се записват два символа (самият символ, който се въвежда, и символа за край на въвеждането) – например

+	\n
---	----
- `getchar` / `scanf("%c", ...)` чете само един символ, което означава, че след прочитането му, буфера няма да е празен –

\n

- Следващо четене на символ няма да изчака вход от потребителя, а ще прочете вече намиращия се в буфера символ - `\n`
- Може да срещнете, че това може да се избегне чрез изчистване на буфера с извикване на `fflush(stdin)`, но това **не е** стандартна функционалност и може да не работи – най-сигурното е при четене празните символи (интервал, нов ред) да се филтрират (пропускат)