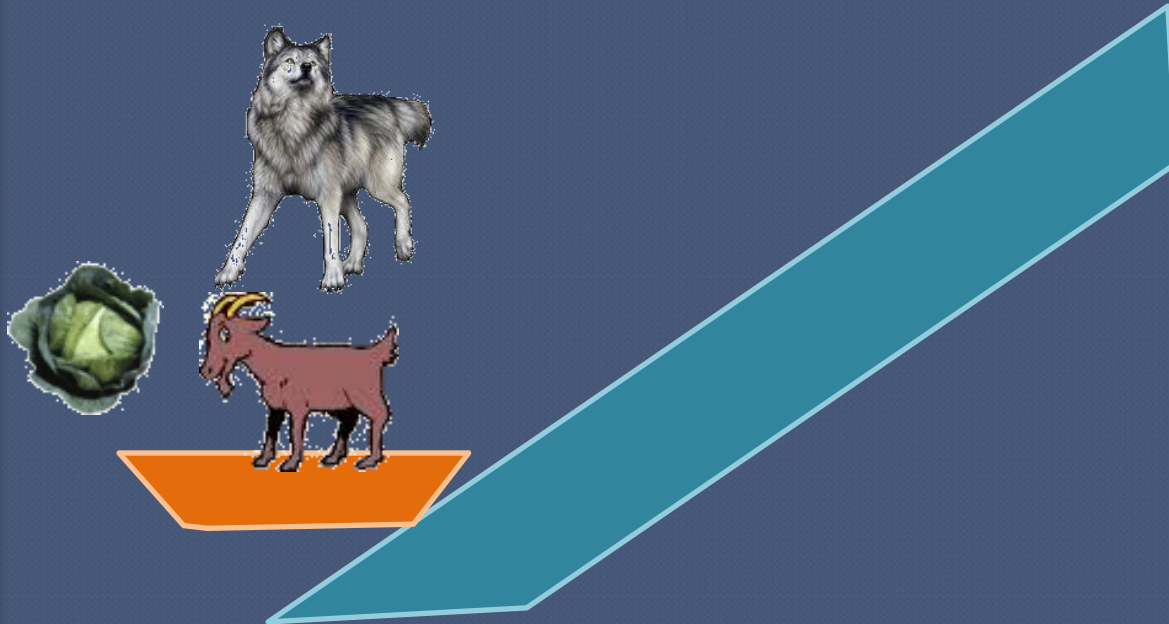


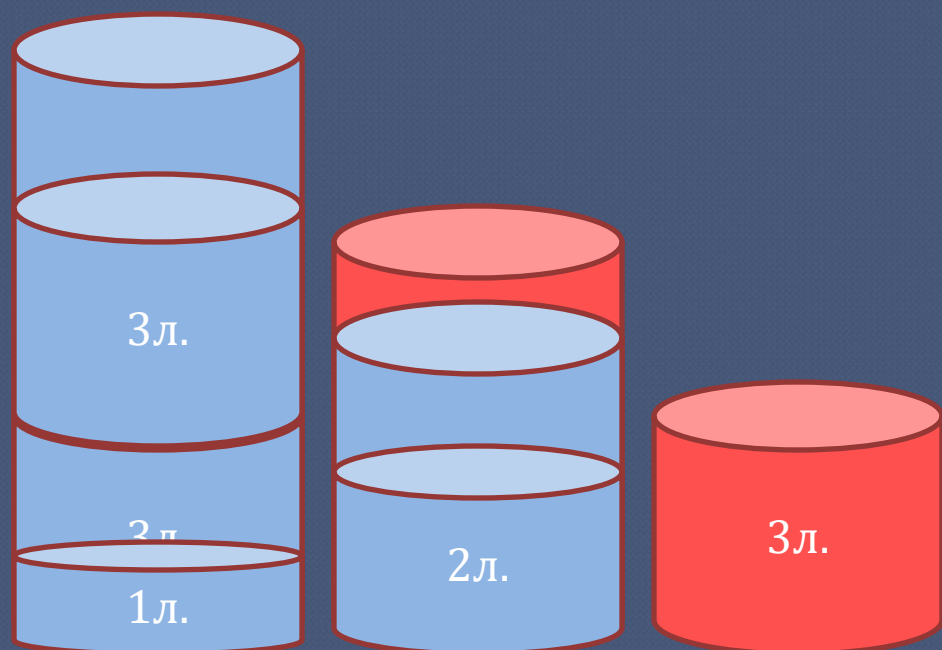
Описание на алгоритми

Известни алгоритмични задачи:

1. Вълк, коза и зеле да преминат от другата страна на реката с 1 лодка.
 - В лодката може да се вози само 1 обект.
 - На един бряг не могат да седят ВК, КЗ



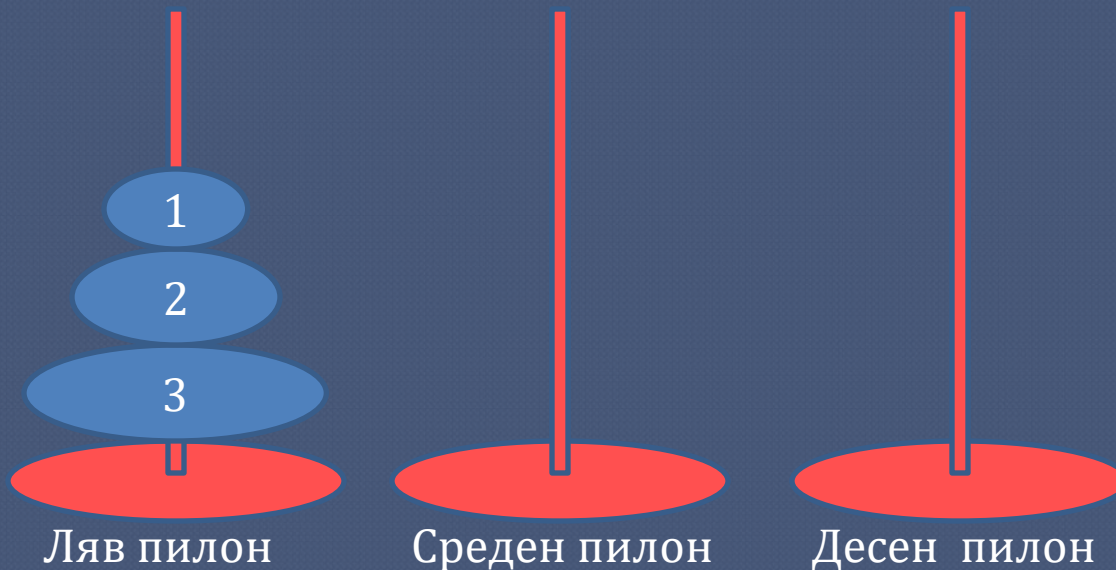
2. Имаме три съда с вместимост 8, 5 и 3 литра.
Съдът от 8л. е пълен с вода.
Разделете водата на 2 равни части.



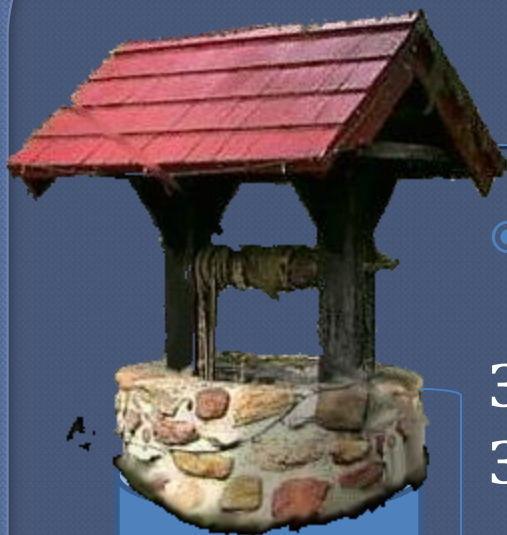
- 1) 8 0 0
- 2) 3 5 0
- 3) 3 2 3
- 4) 6 2 0
- 5) 6 0 2
- 6) 1 5 2
- 7) 1 4 3
- 8) 4 4 0

Задача за ханойските кули:

- На 3 пилонa има 3 диска.
- Прехвърлете ги от левия на десния пилон.
- Можете да поставяте само по-малък диск върху по-голям диск.



Охлюв в кладенец:



- Охлюв се катери от дъното на кладенец дълбок 6 м - нагоре.

За 1 денонощие минава 2м нагоре и 1м надолу.
За колко денонощия ще се изкачи?

6 м.

1

1

1

1

1

- Колко пъти се повтарят действията изкачване 2м и слизване 1м?
- Колко пъти ще се повторят ако кладенеца е дълбок 12м?



Решение

1. Дядото взема овцата и я пренася на другия бряг
2. Връща се сам.
3. Пренася и вълка на другия бряг.
4. Взема овцата и я връща на първия бряг.
5. Взема сламата и я пренася при вълка.
6. Връща се сам.
7. Взема овцата и двамата се прехвърлят на другия бряг.
8. Така четиримата преминаха на другия бряг живи и здрави. 😊

9. ЗАДАЧА

Даден е следният програмен фрагмент, описан чрез псевдокод. Кое число ще бъде изведено в резултат от изпълнението на фрагмента, ако последователно са въведени числата 5, 2 и 9?

Въведи **a**;

Въведи **b**;

Въведи **c**;

d := **a**;

Ако **b** < **d**, тогава

d := **b**;

Ако **c** < **d**, тогава

d := **c**;

Изведи **d**;

Въведете от клавиатурата в полето за отговори стойността на променливата **d**.

Даден е следният програмен фрагмент, описан чрез псевдокод. Кое число ще бъде изведено в резултат от изпълнението на фрагмента, ако последователно са въведени числата 7, 5 и 3?

(3 Points)

Въведи a ; Въведи b ; Въведи c ;

$d := -a + b$;

Ако $b > -d$, тогава $d := -3 + b - a$;

Ако $d < c$, тогава $d := c$;

Изведи d ;

1. Произход на думата алгоритъм

- Мохамед ибн Муса ал Хорезми
- Арабски математик описал
 - представянето на числата в 10 -ична бройна система
 - извършването на аритметичните операции с тях



Ал-Хорезми показва на света с книгите си арабските цифри. През XII-XVII век в Европа е имало остра борба между подържниците на римските цифри и тези, предпочитащи арабските. Но в крайна сметка през XVII окончателните победители стават АЛГОРИТМИЦИТЕ.

Какво е алгоритмично мислене?

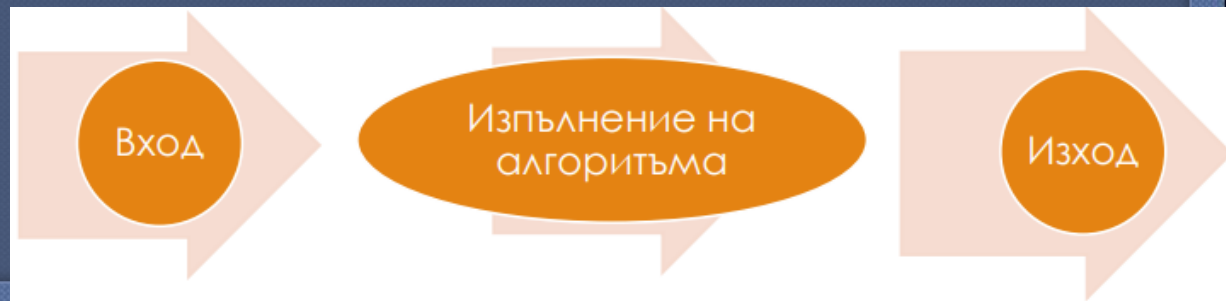
*Това е изкуството да се размишлява,
умението да се планират действията,
способността да се предвиждат
различни обстоятелства и да се
постъпва в съответствие с тях.*

2. Условия при съставяне на алгоритъм:

- ◎ Да представите сложното действие чрез последователност от по-прости действия, достъпни за изпълнителя;
- ◎ Да използвате само елементарни (за изпълнителя) действия;
- ◎ Да опишете ясно и точно последователността, в която трябва да се изпълняват елементарните действия;

3. Определение за алгоритъм:

- а) Елементарно действие е това, което може да се извърши без допълнителни пояснения.
- б) Всяко изпълнение на елементарно действие се нарича стъпка
- в) Алгоритъм е крайна последователност от елементарни действия, след изпълнението на които се достига до решението на дадена задача, или до заключението, че задачата няма решение.



4. Свойства на алгоритмите:

- **Дискретност** - всеки алгоритъм трябва да се състои от краен брой елементарни действия
- **Яснота** - всяка текуща стъпка трябва да е точно указана и да бъде еднозначно определено коя е следващата.
- **Определеност** - при всяко изпълнение с едни и същи входни данни да се достига до един и същ резултат

- ◎ **Масовост**- всеки алгоритъм да се прилага успешно за решаване на голям брой задачи от един и същи тип спрямо различни входни данни
- ◎ **Крайност**- винаги достига до край, независимо от входните данни
- ◎ **Резултатност**- винаги има резултат, независимо от входните данни
- ◎ **Ефективност**- рационалност, влияе върху бързината на изпълнение

4. Начини за описание на алгоритми

- ◎ Словесно
- ◎ Метаезик
- ◎ Блок - схема
- ◎ Програмен език

Задача: Да се състави програма, която решава линейно уравнение от първа степен с едно неизвестно.

Словесно описание

- Описват се словесно последователностите от стъпки, които трябва да се изпълнят за да се реши задачата.
- Всички стъпки са номерирани според реда на тяхното изпълнение.
- Ако се налага нарушаване на този ред (за реализация на разклонения или цикли) се описва изрично коя е следващата стъпка.

Словесно описание на алгоритъм за решаване на линейно уравнение

1. Въвеждаме в ОП стойности за a и b
2. Проверяваме дали $a=0$
3. Ако $a \neq 0$ отиваме на т.7
4. Проверяваме дали $b=0$
5. Ако $b=0$ извеждаме „всяко x е решение” и отиваме на т.9
6. Ако $b \neq 0$ извеждаме „няма решение” и отиваме на т.9
7. Пресмятаме $x = -b/a$
8. Извеждаме стойността на x
9. Край

Чрез метаезик (псевдокод)

Начало

Въведи **a**;

Въведи **b**;

Ако $a=0$ то

Ако $b \neq 0$ то

изведи „няма решение“;

в противен случай

изведи „всяко x е решение“;

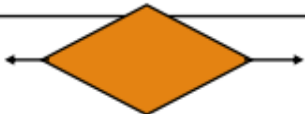
в противен случай

$x := -b/a$;

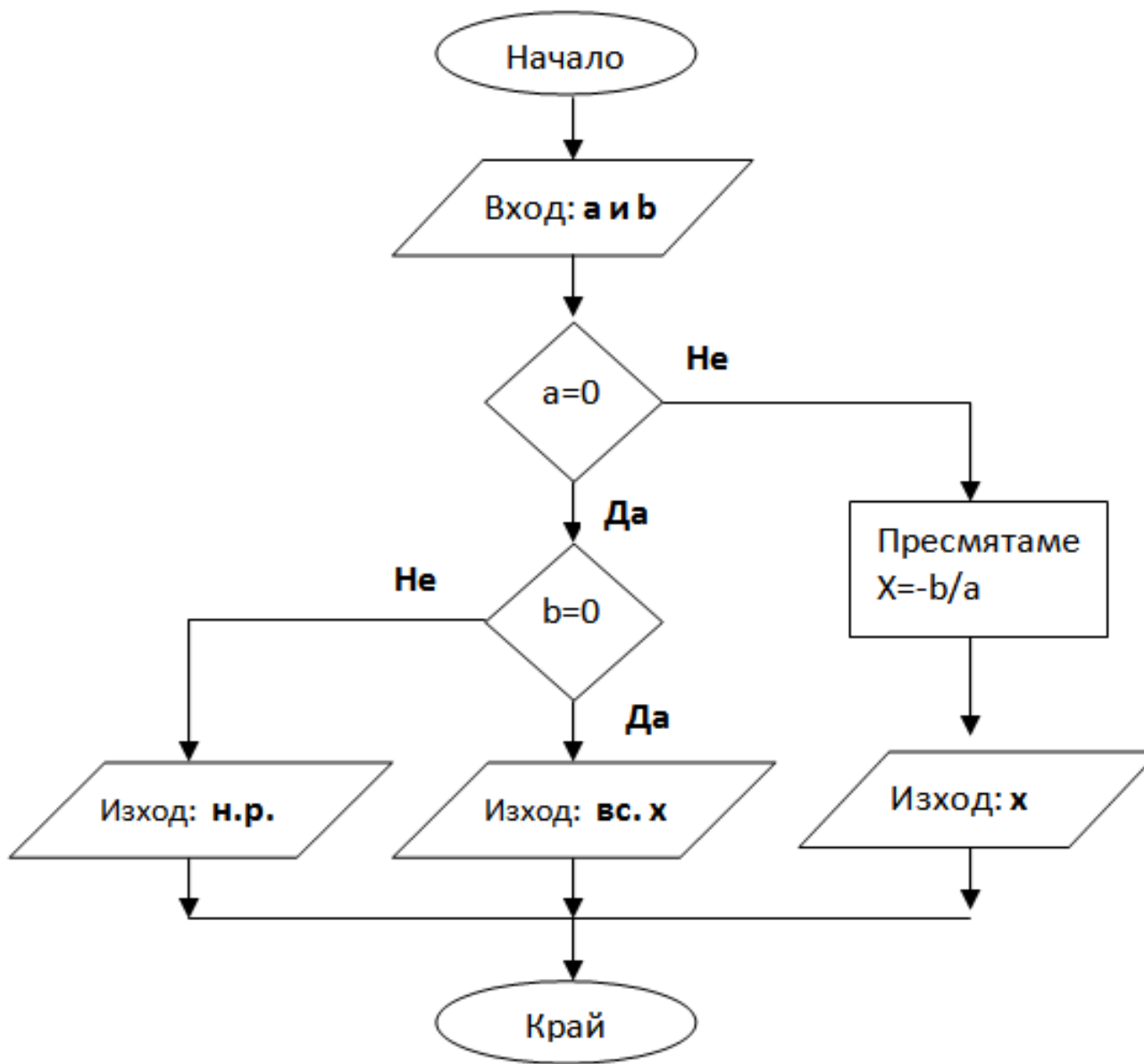
изведи стойността на x ;

Край

Чрез блоксхема

Име на блока	Форма	Пояснение
Начален блок	 начало	Посочва от кой блок започва изпълн. на алг.
Краен блок	 край	Посочва завършване на изпълнението на алг.
Блок за въвеждане	 вход	Съдържа имена на величини, които получават стойност
Блок за извеждане	 изход	Съдържа имена на величини, които се извеждат
Безусловен блок		Съдържа едно или повече елементарни действия
Условен блок		Служи за разклоняване на алгоритъма
Блок за обр. към подалг.		Съдържа име и фактически параметри на подалгоритъм

Чрез блоксхема



Чрез програмен език

```
var a, b, x : real;
```

```
begin
```

```
  write('Enter a= ');
```

```
  readln(a);
```

```
  write('Enter b= ');
```

```
  readln(b);
```

```
  if a=0 then
```

```
    if b=0 then write('All x')
```

```
    else write('No x')
```

```
  else
```

```
    begin
```

```
      x := -b/a;
```

```
      write('x= ', x)
```

```
    end
```

```
end.
```

Pascal

```
#include<iostream.h>
```

```
void main()
```

```
{
```

```
  double a, b, x;
```

```
  cout << "Enter a= ";
```

```
  cin >> a;
```

```
  cout << "Enter b= ";
```

```
  cin >> b;
```

```
  if (a==0)
```

```
    if (b==0) cout << "All x";
```

```
    else cout << "No x";
```

```
  else {
```

```
    x = -b/a;
```

```
    cout << "x= " << x;
```

```
  }
```

```
}
```

C++

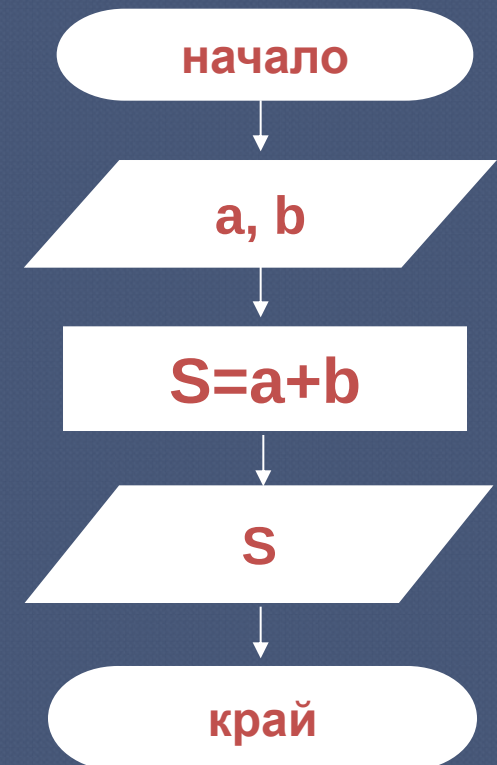

```
static void Main(string[] args)
{
    double a, b, x;
    Console.Write("Въведете a:");
    a = Convert.ToDouble(Console.ReadLine());
    Console.Write("Въведете b:");
    b = Convert.ToDouble(Console.ReadLine());
    if (a == 0)
        if (b == 0) Console.WriteLine("All x");
        else Console.WriteLine("No x");
    else
    {
        x = -b / a;
        Console.WriteLine("x= " + x);
    }
}
```


5. Видове алгоритми според начина на изпълнение

- ◎ Линейни
- ◎ Разклонени
- ◎ Циклични
- ◎ Рекурсивни

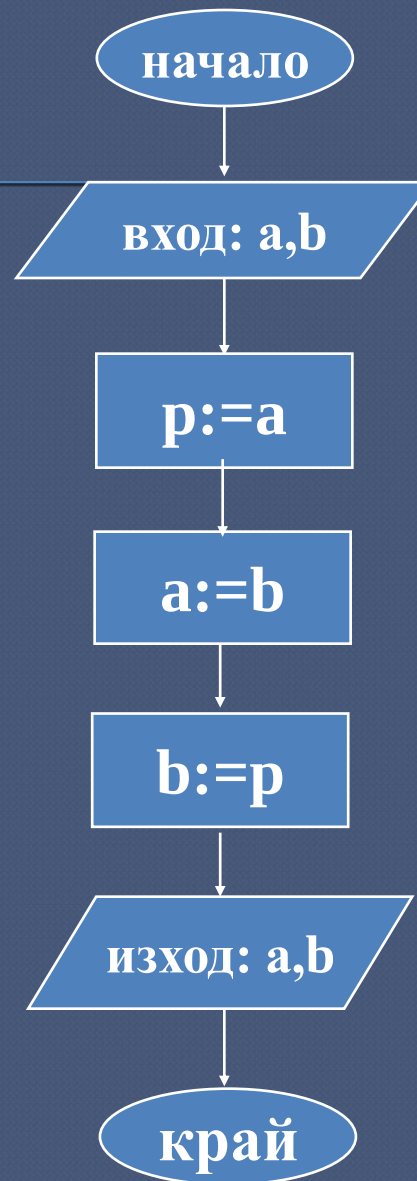
• Линейни алгоритми

- Алгоритми, при които всички действия се изпълняват последователно.



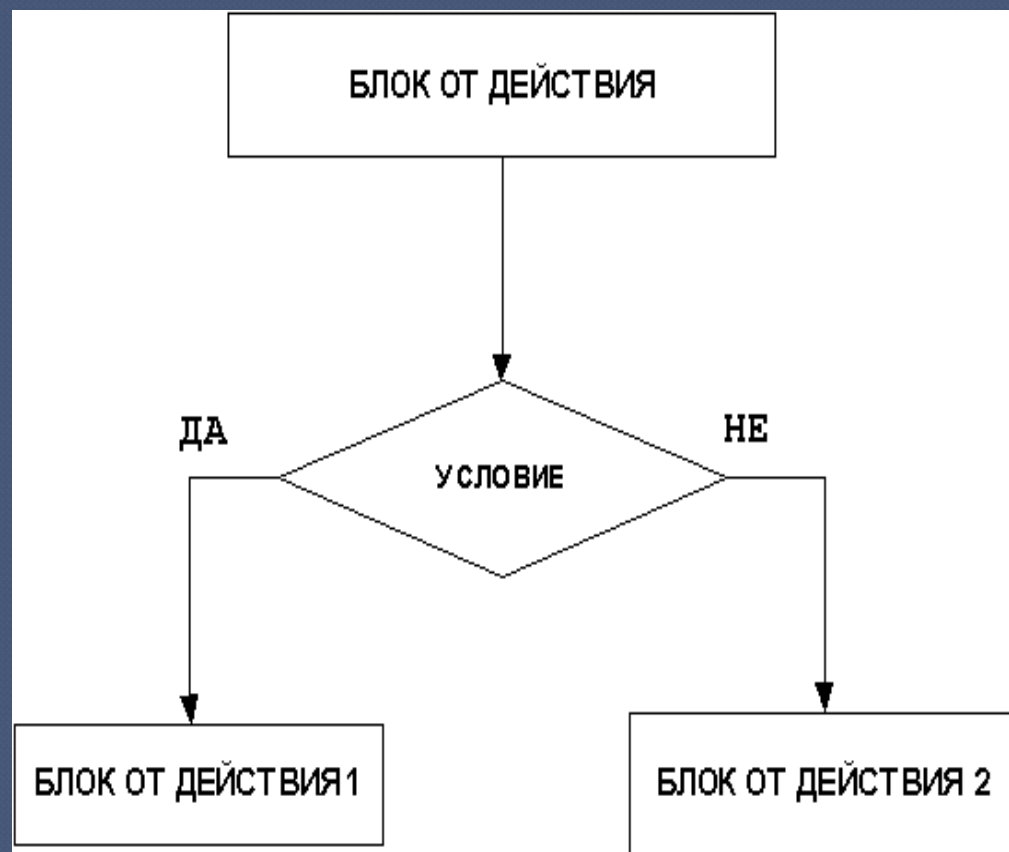
Пример: Линейна блок схема

Размяна
стойностите на
2 променливи



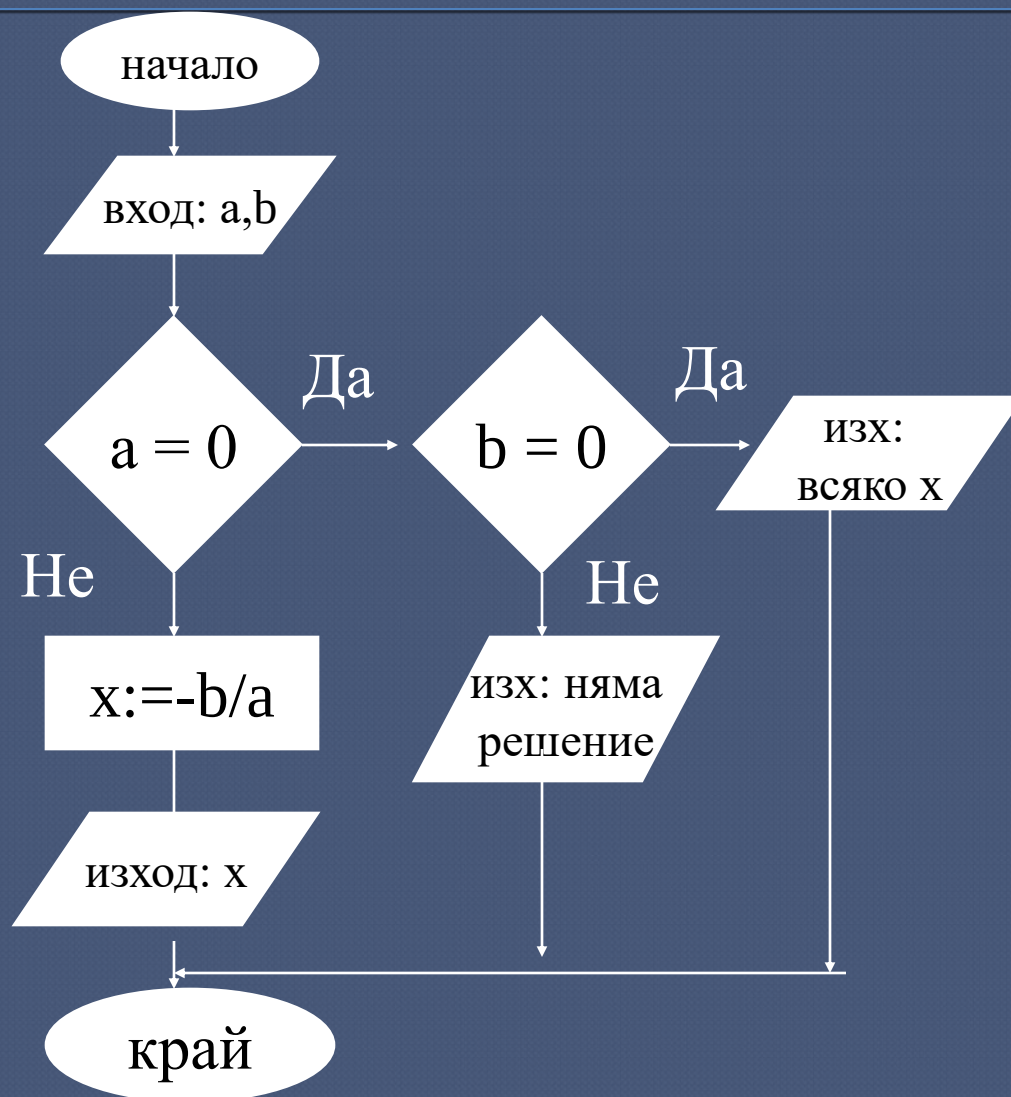
• Разклонени алгоритми

- Алгоритми, при които реда на изпълнение на командите зависи от някакво условие.



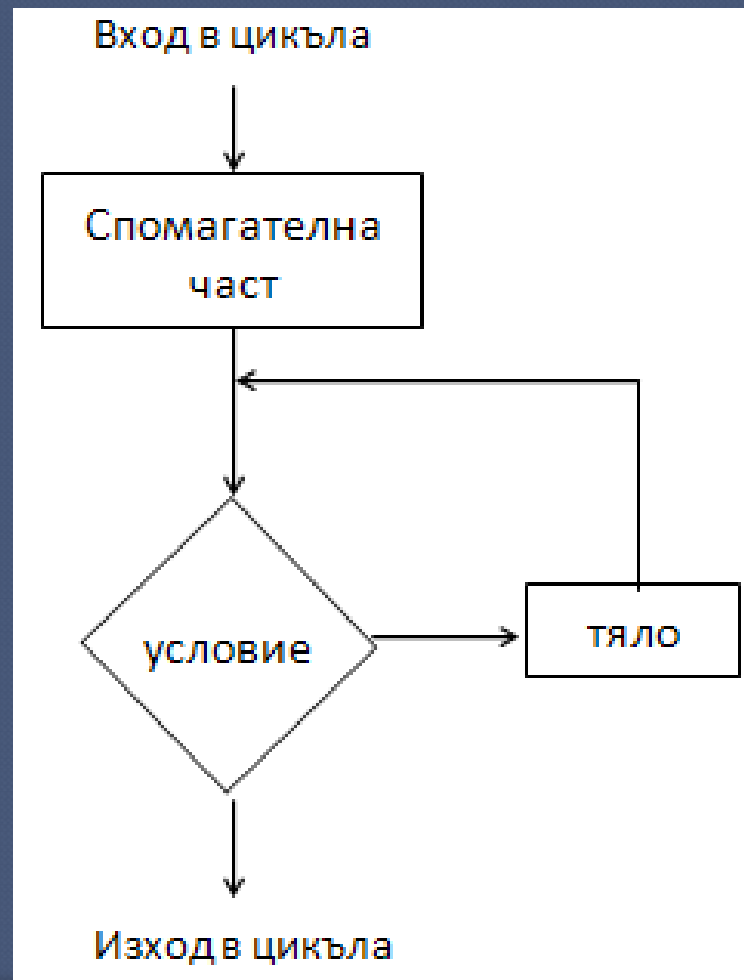
Разклонени блок-схеми

Пример: Решаване на уравнението $ax + b = 0$



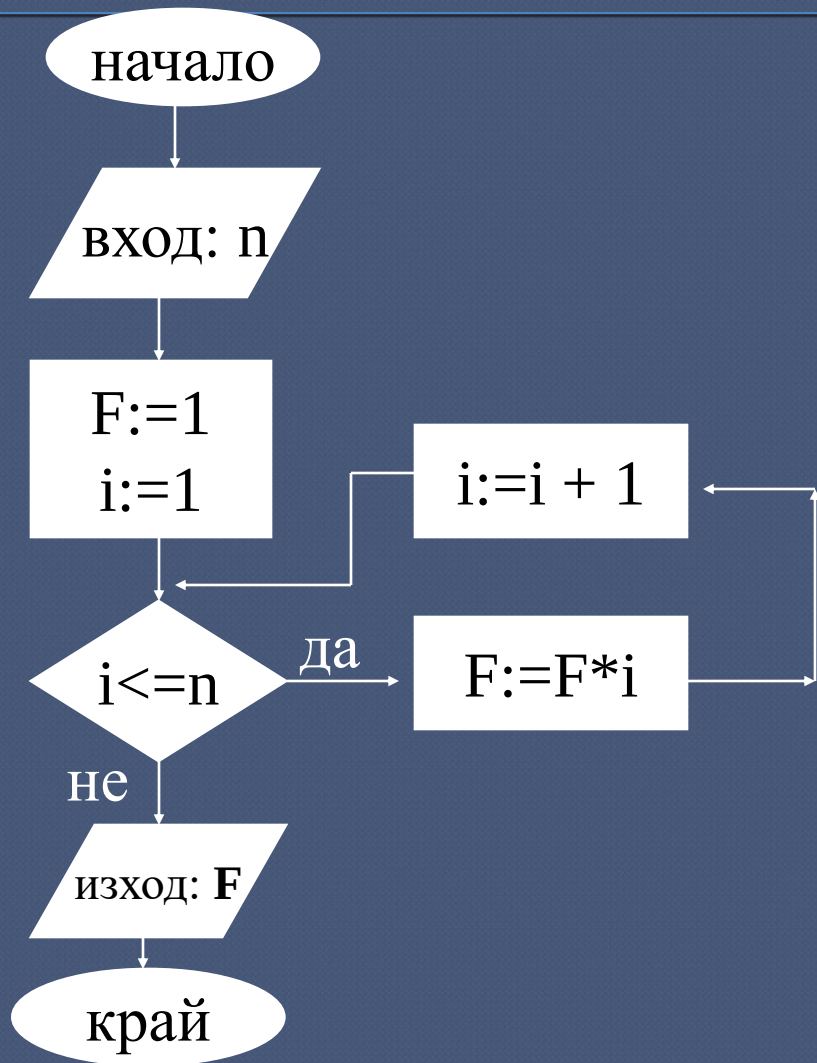
• Циклични алгоритми

- Алгоритми, при които една или повече команди се повтарят многократно до удовлетворяване на някакво условие.



Циклични блок-схеми

Пресмятане на произведението $F=1.2.3.4...n$



Цикли с:

- Предусловие
- Постусловие

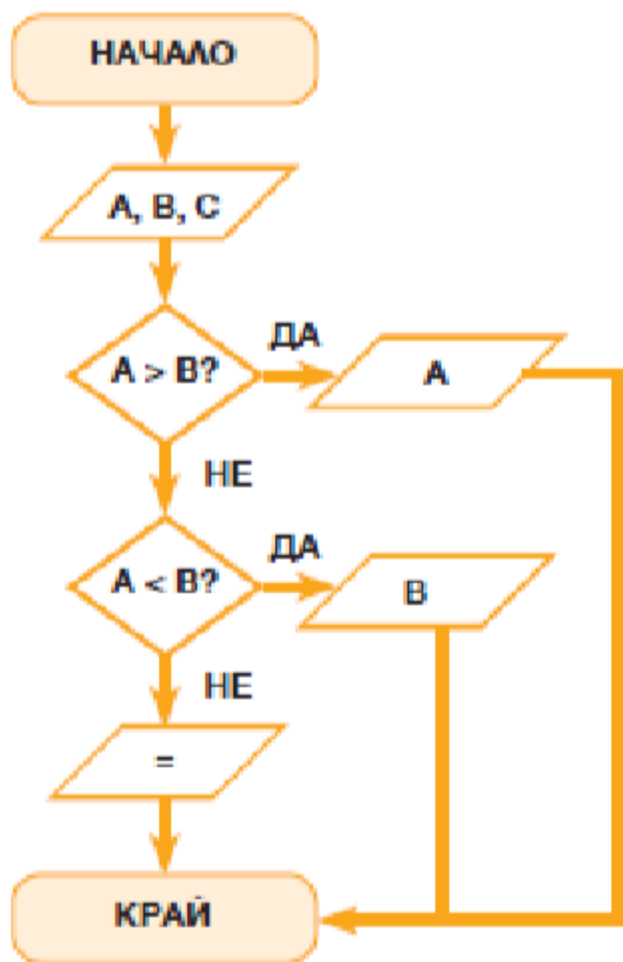
Цикъл с предусловие



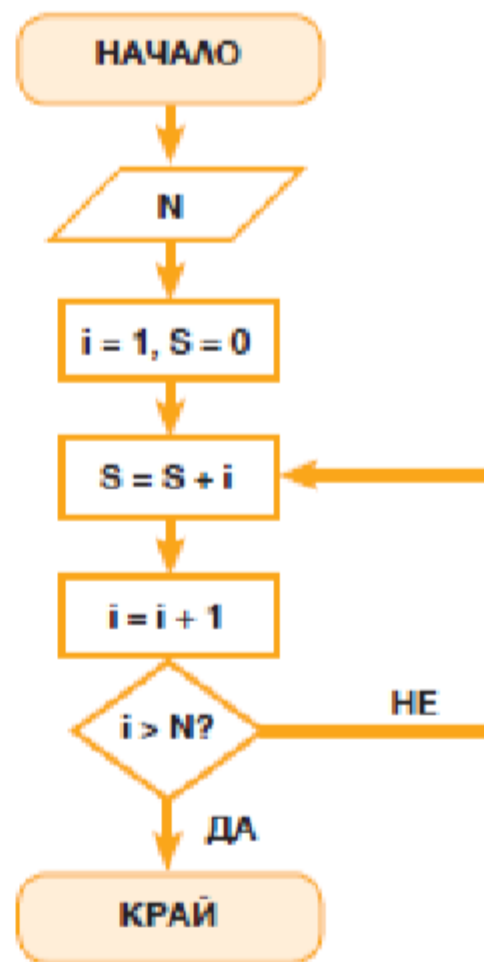
Цикъл с постусловие



Определете резултатите по зададените входни данни



Фиг. 7. $a = -5, b = 2; a = 7, b = 7$



Фиг. 8. $N = 5; N = 100$

Какво ще се получи в резултат на изпълнението на инструкциите в посочения код, ако на променливите a , b , c и d са присвоени съответно стойностите 23, 11, 4 и 825?

Въведи b :

Въведи c :

Въведи d :

$p := a$:

Ако $b > p$, тогава $p := b$;

Ако $c > p$, тогава $p := c$;

Ако $d > p$, тогава $p := d$;

Изведи p ;

Командите за управление на робот са: движение с брой стъпки или направление на движението. Например: **НАПРЕД 10** означава да се придвижи напред 10 стъпки, **НАЗАД 5** – прави 5 стъпки назад, **НАЛЯВО 90** - обръща се наляво на 90°, **НАДЯСНО 90** – обръща се надясно на 90°. Написани са следните програми:

НАПРЕД 10
НАЛЯВО 90
НАПРЕД 10
НАПРЕД 10
НАДЯСНО 90
НАПРЕД 10
НАДЯСНО 90
НАПРЕД 10
НАДЯСНО 90
НАПРЕД 20
НАДЯСНО 90
НАПРЕД 10

НАПРЕД 10
НАЛЯВО 90
НАПРЕД 10
НАЛЯВО 90
НАПРЕД 10
НАДЯСНО 90
НАПРЕД 10
НАДЯСНО 90
НАПРЕД 20
НАДЯСНО 90
НАПРЕД 10
НАДЯСНО 90
НАПРЕД 10

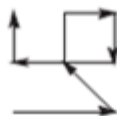
Каква следа е оставил роботът, ако в началото е бил обърнат така, както е на рисунката?



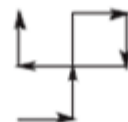
А)



Б)



В)



Г)



Визуален редактор – draw.io

Untitled Diagram.drawio

File Edit View Arrange Extras Help [Last change 2 minutes ago](#)

Master Builder  [Share](#)

100%                

Search Shapes 

Scratchpad   

Drag elements here

General

Text               

Misc

Advanced

Basic

Arrows

Flowchart

Style Text Arrange



☒ Fill 

☐ Gradient

☒ Line 

 1 pt

Perimeter  0 pt

Opacity  100 %

☐ Shadow

Edit Style

Copy Style Paste Style

Set as Default Style

Property Value

