Лабораторно упражнение № 2

Тема: Алгоритмизация на числени и логически задачи. Документиране.

I. Цел на лабораторното упражнение

Усвояване на знания и умения при съставяне на линейни, разклонени и циклични алгоритми.

II. Постановка на задачата

В настоящото лабораторно упражнение ще бъдат разгледани основните методи за алгоритмизиране на програмни задачи. Ще бъдат разгледано описанието на алгоритми посредством блок схеми (линейни, разклонени и циклични алгоритми).

III. Теоретични сведения

Използването на компютри за автоматизация на обработката на информация и управление в която и да е човешка дейност се основава на принципа на програмното управление. В основата на компютърното управление е програмата. Всяка програма се пише от човек, но преди това е необходимо да се опише пълно, ясно и еднозначно изчислителният процес. Това еднозначно описание на процеса представлява алгоритмът на действието и.

Под алгоритъм се разбира: **Точно упътване за решаване на един проблем или определен вид проблеми. Той се състои от крайна поредица от указания, които се изпълняват едно след друго и отчасти повтарят в определен ред.**

В зависимост от своята структура алгоритмите могат да бъдат:

- **Линейни** – При тях последователността от действия винаги е една и съща. Всяка стъпка на алгоритъма има само една предходна и една следваща.
- **Разклонени** – Последователността от извършвани действия зависи от стойността на входните данни. Тези алгоритми съдържат команди, при които в зависимост от изпълнението или не на дадено условие, се определят следващите за изпълнение команди
- **Циклични** – Определена част от действията се повтаря многократно докато не се изпълни зададеното условие.

Най-често се срещат комбинираните алгоритми при които отделни части отговарят на линейни, разклонени или циклични.

Средства за описване на алгоритми.

За представяне на алгоритмите се използват различни езици: естествени, алгоритмични, параметрични, графични и др., които се характеризират с различна степен на формализация на описанието, удобство за запис и четене, възможност за обмен и т.н. Всяка програма е специфично представяне на алгоритъма, разбира се, като се пренебрегнат ограниченията, налагани от съответния език за програмиране.

а) Словесно описание

Словесното описание представлява набор от указания, в които чрез думи от някакъв естествен език са посочени действията, които трябва да бъдат извършени. Възможно е при описанието да се използват означения от съответната предметна област. Този начин на описание не е много популярен, тъй като за много думи от естествените езици съществува неоднозначно тълкуване.

б) описание чрез език за програмиране (псевдокод)

Алгоритмите се описват като се използва ограничено множество от конструкции на естествен език. Множеството от конструкции се разглежда като език за проектиране на алгоритми.

Пример: Алгоритъм на Евклид чрез псевдокод

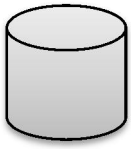
```
Въведи A, B
Докато A ≠ B повтаряй
   Ако A>B, то A:=A-B, иначе B:=B-A;
Изведи A
Край.
```

в) Графично описание посредством блок-схеми;

Блок схемата е графично изразно средство, за описание на алгоритми, което дава възможност ясно да се покажат връзките между отделните инструкции.

Таблица 1: Стандартни символи използвани за описание на блокови алгоритми

Наименование	Символ	Изпълнявана функция
1. Изчислителен блок		Изпълнява изчислително действие или група от действия
2. Логически блок		Избор на направления за изпълнение на алгоритма в зависимости от условието
3. Блокове за въвеждане и извеждане на информация		Въвеждане или извеждане на данни в зависимост от физическия носител
		Извеждане на данни на печатащо устройство

		Ръчно въвеждане на данни
		Дисплей
		Четене и запис от хард диск
4. Начало/край		Начало или край на програмата или подпрограма
5. Подпрограма		Изпълнение на стандартна или потребителска подпрограма
6. Блок модификация		Изпълнение на действия за промяна на точката (пункта) на алгоритъма
7. Вътрешно-страничен съединител		Вътрешно страничен преход
8. Между-страничен съединител		Между-страничен преход (преминаване на нова страница)

Описанието чрез блок-схеми е удобно, тъй като се предлага визуална представа на логическите връзки между отделните действия в алгоритъма. Описанието се извършва от определени символи, наречени блокове, като всеки символ има точно определен смисъл. Блоковете имат вида на геометричните фигури: правоъгълник, ромб, успоредник. Последователността се задава чрез стрелки. Блок-схемата започва с начален блок, в който се записва указанието, с което започва алгоритъма. В таблица 1 са показани по-важните приети у нас по стандарт символи използвани в блоковите схеми (БДС 19103—78).

Основни видове алгоритми

Линейни алгоритми

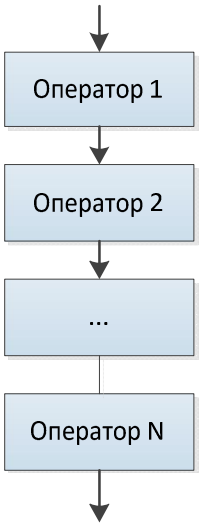
При тях последователността от действия винаги е една и съща. Всяка стъпка на алгоритъма има само една предходна и една следваща (фиг. 2).

Пример:

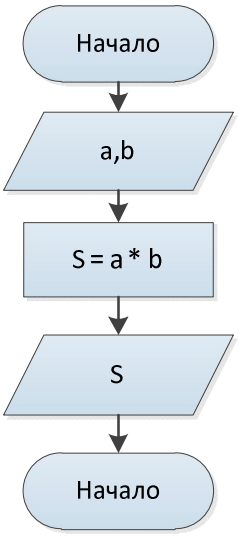
Нека да разгледаме алгоритъма за пресмятане лицето на правоъгълник по формулата $S = a * b$.

Стъпка 1: Начало.
Стъпка 2: Въведете и запомнете стойностите на a и b.
Стъпка 3: На променливата S присвояваме произведението на $a * b$
Стъпка 4: Изведете стойността на S като резултат.
Стъпка 5: Край.

Блоковата схема на алгоритъма е дадена на фиг. 3



Фиг. 2. Блокова схема на линеен алгоритъм



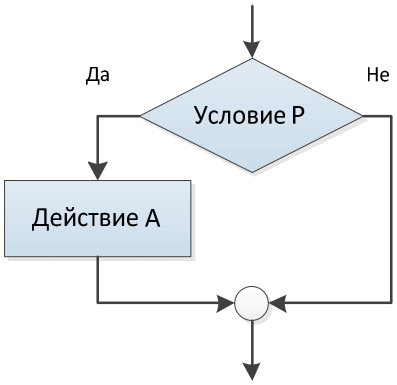
Фиг. 3. Блокова схема на алгоритъм за решаване на формулата $S = a * b$

Разклонени алгоритми

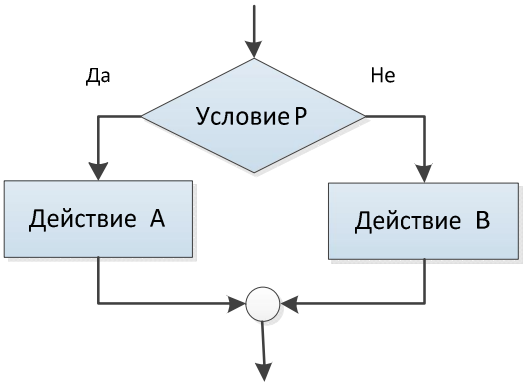
Съществуват две форми на разклонен алгоритъм:

Кратка – Използва се тогава, когато е необходимо да се изпълни действието А само при определено условие Р. Ако условието не е изпълнено, то командата не се изпълнява.

Пълна – Ако се изпълни условието Р, се извършва действие А, а ако не се изпълни действие В.



Фиг. 4. Условен оператор в непълен формат



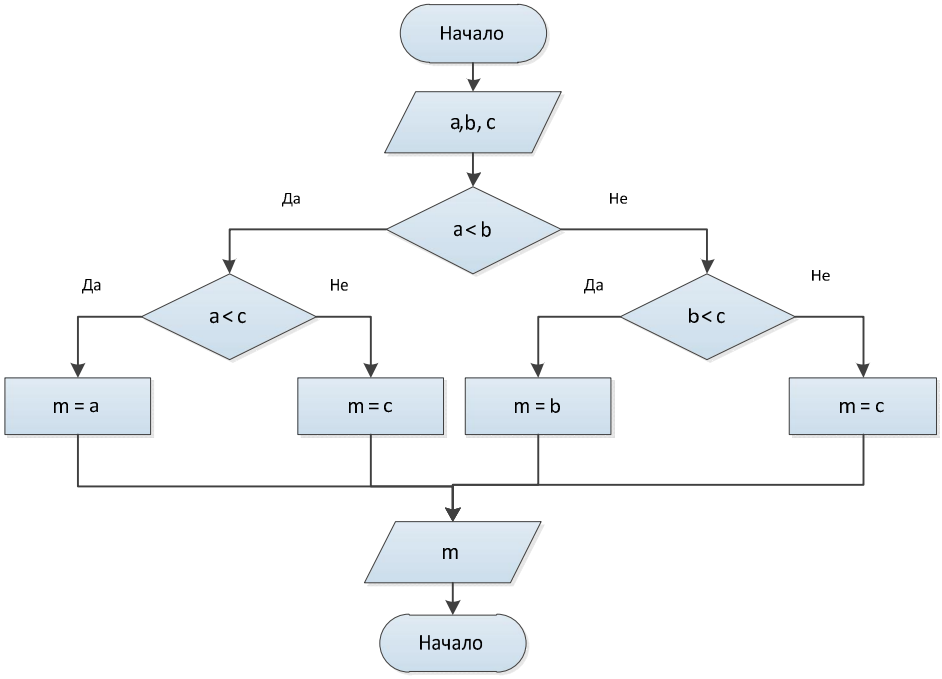
Фиг. 5. Условен оператор в пълен формат

Пример: Да се синтезира алгоритъм за намиране на най-малкото от три числа.

Така поставената задача може да бъде решена по два начина. При първият се изпълняват следните стъпки:

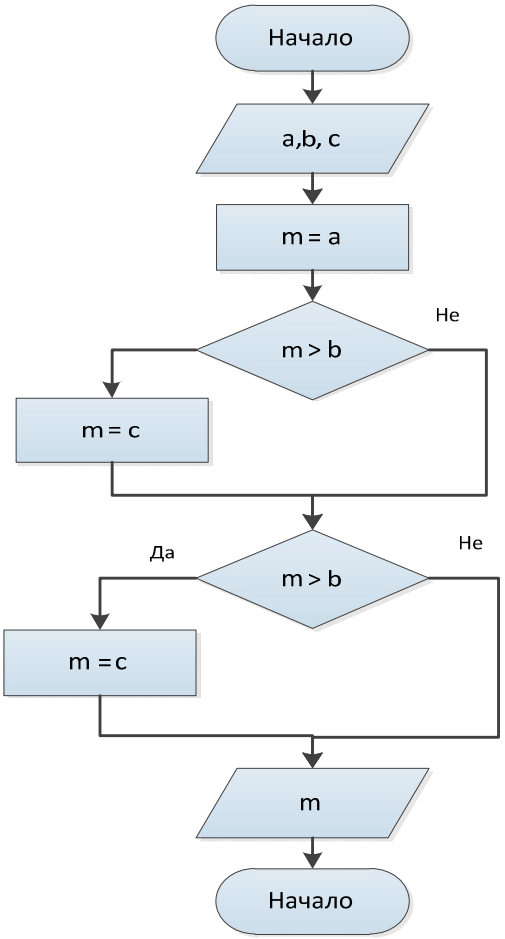
Стъпка 1: Начало.
Стъпка 2: Въведете и запомнете стойностите на a, b и c.
Стъпка 3: Проверете дали a < b
Стъпка 4: Ако a < b, проверете дали a<c
Стъпка 5: Ако a<c на m присвоете стойността на a, ако не е присвоете на m стойността на c
Стъпка 6: Ако a ≥ b, проверете дали b<c
Стъпка 7: Ако b<c на m присвоете стойността на b, ако не е присвоете на m стойността на c
Стъпка 8: Изведете стойността на m като резултат.
Стъпка 9: Край.

Алгоритмът е даден на фиг. 6



Фиг.6. Алгоритъм на програма за намиране на най-малкото от 3 числа

Втори вариант на алгоритъма е показан на фиг. 7. Опитайте се да запишете стъпките за изпълнението му.



Фиг.7. Алгоритъм на програма за намиране на най-малкото от 3 числа – втори вариант

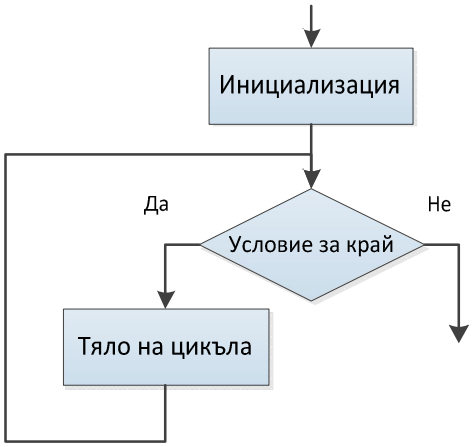
Циклични алгоритми

Много често се налага определени действия да бъдат повторени многократно докато не се изпълни зададеното условие. Групата от блокове, които са описани веднъж, но се изпълняват многократно се наричат ТЯЛО НА ЦИКЪЛА. При тях задължително има условен блок. А необходимо изискване е да се осигури край на цикличните повторения (излизане от цикъла). В тялото на цикъла, трябва да се променя стойността поне на 1 променлива участваща в условието за край. В противен случай ще се получи безкраен цикъл.

Циклите биват 3 вида:

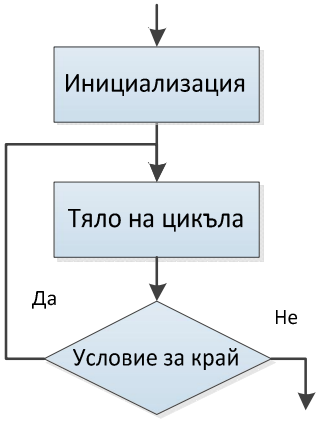
- Цикъл с предусловие
- Цикъл със следусловие (постусловие
- Цикъл с параметри

1. **Цикъл с предусловие** – при тях първо се проверява условието и ако то е изпълнено се изпълняват операциите от тялото на цикъла. Ако не е изпълнено още в началото е възможно тялото на цикъла да не се изпълни нито веднъж.



Фиг. 8 Цикъл с предусловие

2. **Цикъл със следусловие (постусловие)** – при тях първо се изпълняват операциите от тялото на цикъла и след това се проверява условието за продължаване на цикъла. Ако то е изпълнено се връща за повторно изпълнение на тялото, ако не е се излиза от цикъла.



Фиг. 9 Цикъл със следусловие

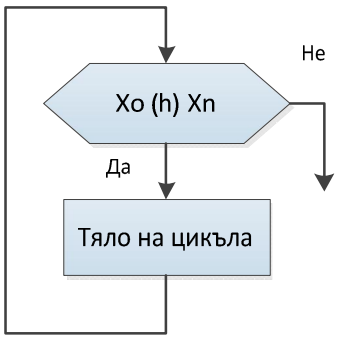
3. **Цикъл с параметри,** или цикъл с брояч или аритметичен цикъл- това е цикъл с предварително зададен брой повторения.

В блока за условие се задава закона за промяна на параметъра.

Xo - първоначалната стойност на бояча

h– (индексна променлива) брояч

Xn – последна стойност на брояча



Фиг. 10 Цикъл с параметри

При използването на цикли с броячи трябва да се спазват следните условия:

- Началната и крайната стойност на брояча трябва да са от един и същ тип.
- Забранява се в тялото на цикъла да се променя значението на началната стойност.
- Ако началната стойност на брояча е по-голяма от крайната стъпката на нарастване е отрицателна.
- След като излезете от тялото на цикъла индексната променлива е неопределена и неможе да се използва.

Всеки един цикъл има 4 задължителни елемента:

- Установяване на началното състояние (началните стойности);
- Тяло – група от оператори, които се изпълняват многократно;
- Условие за изход от цикъла;
- Обновяване на условието за излизане от цикъла (брояч).

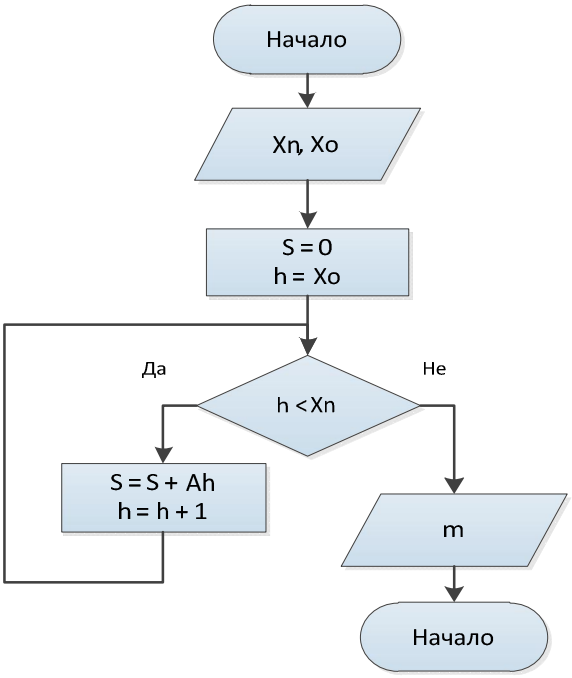
Пример:

Пример за цикъл е задачата за намиране сумата на редица от числа. Сумиране на голям брой числа се използва при обработката на различна статистическа информация или при анализа на различни динамични процеси.

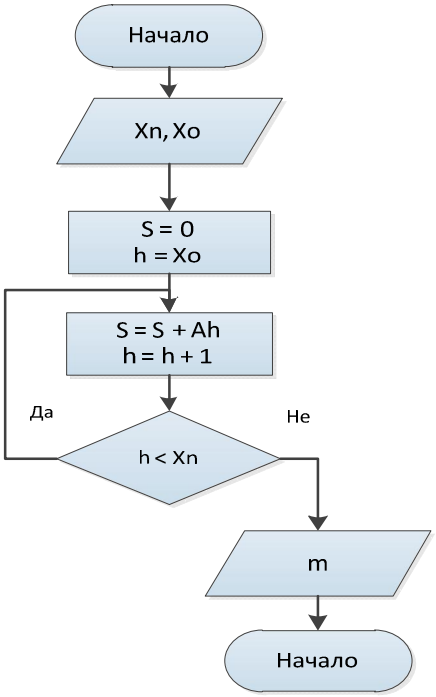
В примера индексната променлива ще бъде зададена като h. Началната стойност като Xo. Крайната с Xn, а сумата със S.

В алгоритъма първо се въвеждат стойностите на Xo и Xn. Задава се начална стойност 0 на S. На брояча h се задава стойността на Xo. Проверява се дали стойността на брояча е достигнала крайната стойност Xn. Ако не към стойността на S се добавя стойността на поредния елемент на масива (под масив в програмирането се разбира

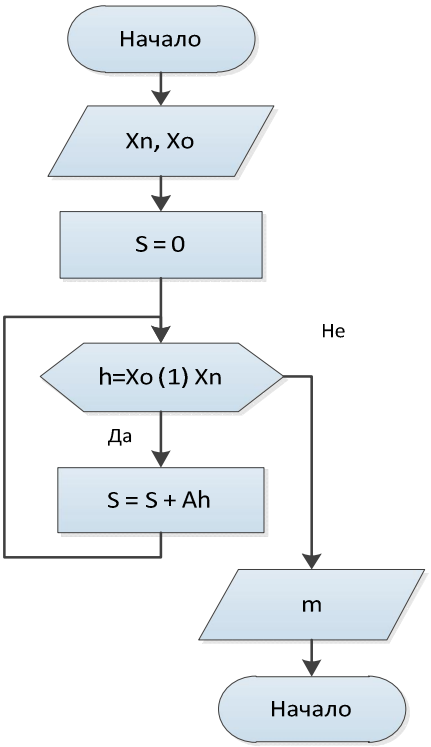
вектор ред от математиката).



Фиг.11. Алгоритъм на програма за намиране сумата на редица от числа реализирана с цикъл с пред условие



Фиг.12. Алгоритъм на програма за намиране сумата на редица от числа реализирана с цикъл със следусловие



Фиг.13. Алгоритъм на програма за намиране сумата на редица от числа реализирана с цикъл с параметри

IV. Задание за работа

Зад 1. Да се състави алгоритъм на програма за намиране на по-малката от две стойности.

Зад 2. Да се състави алгоритъм на програма за решаване на квадратно уравнение.

Зад 3. Да се състави алгоритъм на програма за намиране на n!

Зад 4. Да се състави алгоритъм на програма за въвеждане и извеждане елементите на масив.

Зад 5. Да се състави алгоритъм на програма за изчертаване на „Фигури на Лисажу".

Зад 6. Да се състави алгоритъм на програма за изчертаване на логаритмична спирала.

Зад 7. Да се състави алгоритъм на програма за намираме на производната на функцията $(\ln x)^{(n)}$

Зад 8. Да се състави алгоритъм на програма за търсене на минималната стойност в едномерен масив.

Зад 9. Да се състави алгоритъм на програма за търсене на максималната стойност в матрица.

Зад 10. Да се състави алгоритъм на програма за сортиране на едномерен масив по метода на пряка размяна.

Зад 11 Да се състави алгоритъм на програма за сортиране на масив по метода на пряка селекция.

Зад 12. Поставена е задача за изграждане на сортировъчен автомат за сортиране на гайки с размер ϕ10. при толеранс ±0.1 mm. Първоначално детайлите се намират в един контейнер, след, което се разпределят в други три според размера им. Да се състави алгоритъм за действие на сортировъчния автомат при условие, че бункерите побират по 10000 детайла.

Зад 13. Поставена е задача за изграждане на светофарна уредба на кръстовище при следните условия:

- Време за превключване 1min.
- Време за жълт сигнал 10s.
- Да се издава различен звуков сигнал при разрешени и забранено преминаване.
- Да има цифрова индикация на времето на съответния сигнал.

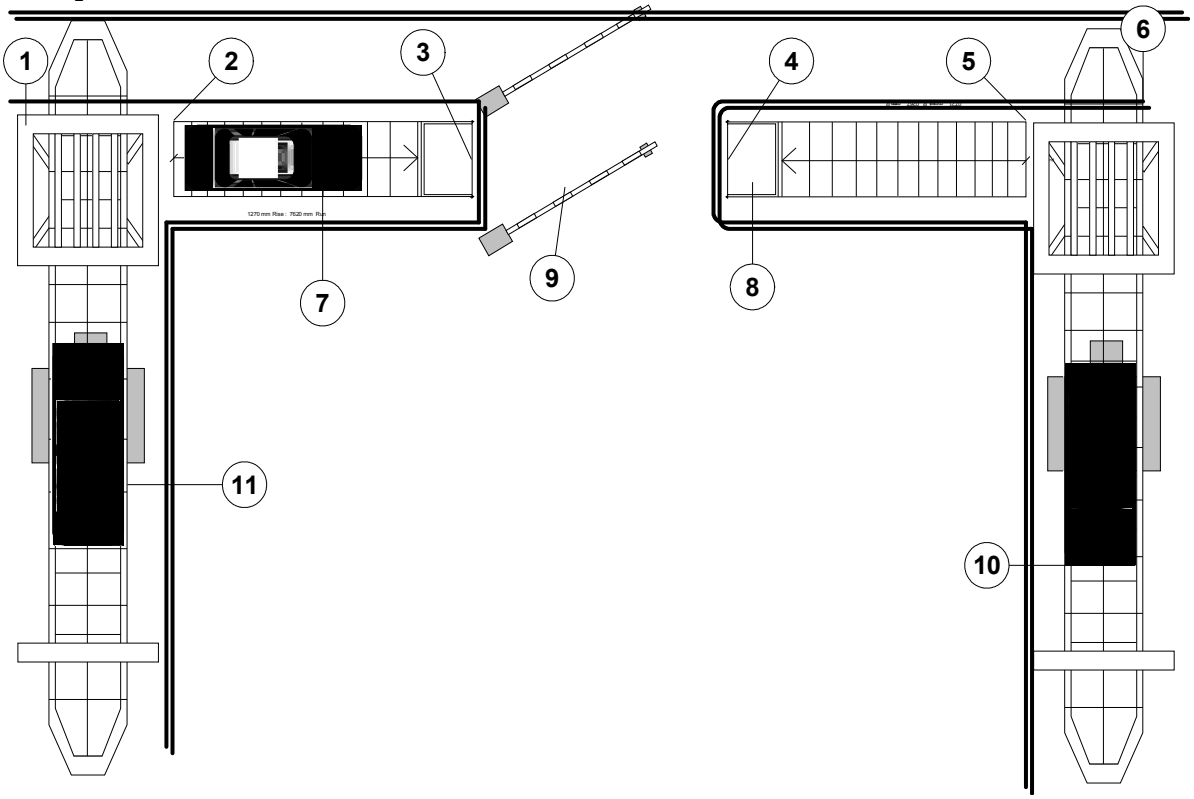
Да се състави алгоритъм на функциониране на светофарна уредба по зададените изисквания.

Зад 14. Поставена е задача за проектиране и изграждане на автоматизирана поточна линия за сглобяване на автомобили. Поточната линия е разположена под формата на буквата П (фиг. 8) с ляв(11) и десен (10) клон.. В горния край на линията е необходимо да се направен прелез охраняван посредством бариери (9), през който да преминават електрокарите с частите за сглобяване. За транспортиране на сглобяваните коли от левия до десния клон на поточната линия се използват две робоколички

(7,8). Посредством сигналите получени от сензори 1 до 6 се следи положението на колите. Действието на поточната линия е следното:

Колата се придвижва по левия клон на поточната линия с постоянна скорост. При достигане края на клона сензор 1 се активира. В резултат на което колата се завърта на 90° и се премества върху робоколичката. След активирането на сензор 2, се издава звуков и светлинен сигнал. Бариерата се спуска, а двете робоколички тръгват една към друга. След активиране на сензори 3 и 4 робоколичките спират и колата се премества от първата на втората количка. Двете колички тръгват към изходно положение. При активиране на сензор 5, количка 2 спира а колата се прехвърля от количката върху поточната линия. При активиране на сензор 6 платформата с колата се завърта на 90° и се подава към транспортната лента.

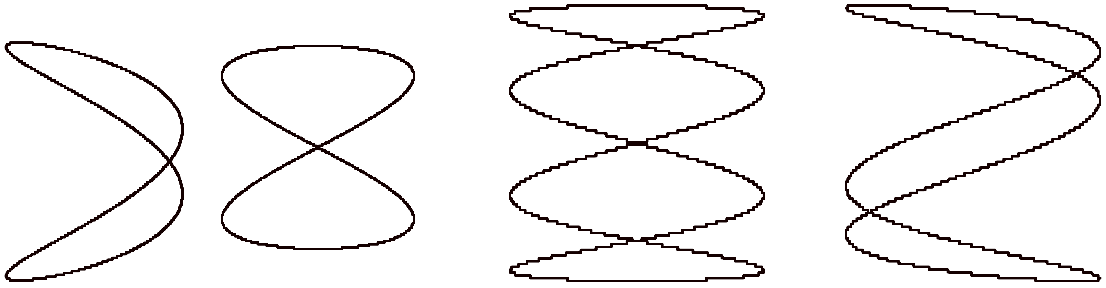
Да се реализира блок схема на алгоритъма на работа на така дефинираната система.



V. Указания за работа

1. Фигурите на Лисажу (Фиг. 9) се получават, когато два електрически сигнала се подадат на двата входа на осцилоскоп. При това на екрана на осцилоскопа се появява фигура с определена форма по която може да се съди за разликата в честотата, големината и фазата на двата сигнала.

Програмно може да се реализира като при изчертаване за X координата се вземе стойността от изчислението на даден хармоничен сигнал (синусоида) , а по оста У се подаде стойността на друг хармоничен сигнал с променена фаза или честота.



x=sin(2t+45) y=sin(t) | x=sin(2t) y=sin(t) | x=sin(4t) y=sin(t) | x=sin(3t+120) y=sin(t)

Фиг. 9

2. Уравнението на логаритмична спирала е $\rho=\alpha.e^{\kappa\varphi}$ к > 0

$$(\ln x)^{(n)}=(-1)^{n-1}.\frac{1.2.3...(n-1)}{x^n}$$

3. Описание на алгоритъма чрез пряка селекция. Масивът A[1],...A[n], се обхожда и се избира елемента с най-малка стойност. След това той разменя мястото си с първият елемен A[1]. Отново масивът се обхожда и се намира с най-малка стойност, след което се разменя с A[2]. Масивът е подреден след (n-1) на брой преминавания.

4. Описание на алгоритъма на пряка размяна (метод на мехурчето) Масивът A[1]…A[n] се обхожда от индекс n до индекс 1, като всеки два съседни елемента се сравняват. Ако десният елемент е по-малък от левия, те сменят местата си. След първото обхождане елементът с най-малка стойност е в най-лявата част на масива. Повторението на тази операция за останалите (n-1) елемента придвижва най-малкия от тях на втора позиция и т.н. След (n-1) на брой обхождания масивът се подрежда.

VI. Контролни въпроси

1. Какво разбирате под термина програмен алгоритъм?
2. Кои алгоритми са линейни?
3. Кои алгоритми са разклонени?
3. Кои алгоритми са циклични?
4. Какво представлява низходящото проектиране на програми?
5. Какво представлява възходящото проектиране на програми?
6. Какво представлява модулното програмиране?
7. Какво е структурно програмиране?
8. Какво разбирате под термина “итерация”?
9. Какво разбирате под термина “рекурсия”?

VII. Литература

1. *Бозм,* Б. У. Инженерное проектирование программного обеспечения, М., Радио и связь, 1985.
2. *Вирт,* Н. Систематическое программирование. Введение. М., Мир, 1977.
3. *Гласс,* Р. Руководство по надежвому программированию. М., Финанси и статистика, 1982.
4. *Гласс,* Р., Р. *Нуазо.* Сопровождение программного обеспечения. М., Мир, 1983.
5. *Грис,* Д. Наука программирования. М-, Мир, 1984.
6. *Дени Ван Тасел.* Стил, ефективност, настройка и тестване в програмирането. С., Техника, 1979.
7. *Лингер,* Р., Х. *Миллс* и *Б. Уипгт.* Теория и практика структурного программирования. М., Мир, 1982.
8. *Майер,* Г. Искусство тестирования программ. М,, Финанси и статистика, 1982.
9. *Тайер,* Т. и др. Надежность программного обеспечения. М-, Мир, 1981.