

Дәріс-2.

Тақырыбы: Негізгі алгоритмдік құрылымдар. Сызықтық және тармақталу алгоритмдері

Жоспар:

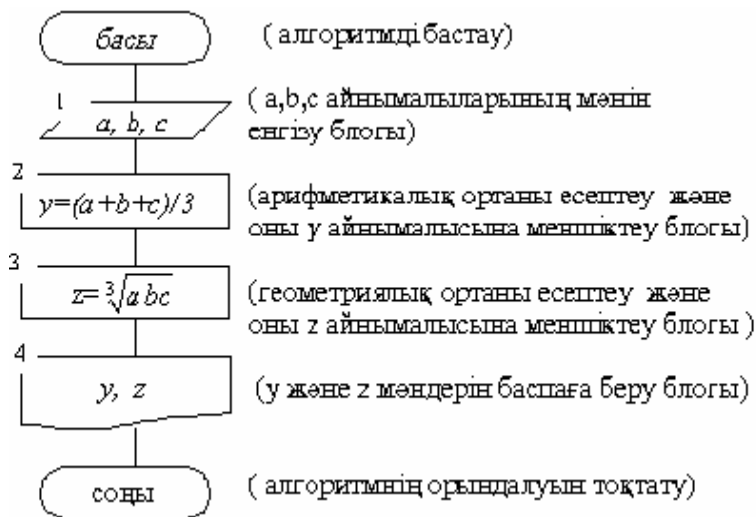
1. Сызықтық алгоритм
2. Тармақталу алгоритмі
3. Қайталану алгоритмі

Алгоритм құру барысында негізгі үш құбылысты - **сызықтық, тармақталу, қайталану** (цикл) негізгі құрылымдар қолданылады. Осыған сай алгоритм *сызықтық, тармақталған* және *қайталанған* құрылымды алгоритм болуы мүмкін. Күрделі мәселелерді шешу алгоритмдері аталған негізгі үш құбылыстың есептің жеке бөліктеріне сәйкес келетін тізбектерінен тұрады.

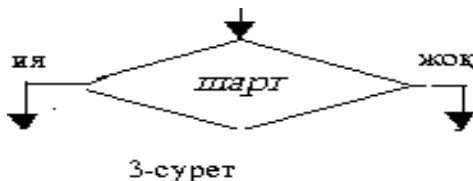
Сызықтық құрылымды алгоритм - әрбір нұсқауы көрсетілген тәртібімен бірінен соң бірі, тізбектеліп орындалатын алгоритм.

1-мысал. a, b, c арқылы өрнектелген үш нақты санның арифметикалық және геометриялық ортасын анықтау керек.

Шешуі: Берілген сандардың арифметикалық ортасын y , геометриялық ортасын z арқылы белгілейік. Онда бұл есепті ЭЕМ-де шешудің қарапайым алгоритмінің блок-схемасын төмендегінше көрсетуге болады:



Ыңғайлы болу үшін әрқашан алгоритмнің басы мен соңын білдіретін

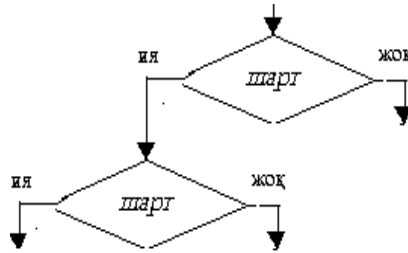


міндетті екі блоктан өзге негізгі блоктар нөмірленген. Блок-схемада көрсетілген 1-4 блоктар бірінен соң бірі, тізбектеліп орындалады, демек бұл - сызықтық алгоритмнің мысалы, мұнда 2-ші және 3-блоктардың

орнын ауыстыруға да болады, бұл есептің дұрыс нәтижесін алуға кері әсерін тигізбейді.

Есепті шешу барысында алдын ала берілген немесе есептеу барысында пайда болған қандай да бір шарттың орындалу/орындалмауына байланысты алгоритмде көрсетілген кейбір нұсқаулар орындалатын/орындалмайтын жағдайлар, яғни *тармақталу құбылысы* жиі кездеседі. Мұндай алгоритмдерді **тармақталған құрылымды алгоритм** деп атайды. Тармақталуды блок-схема тілінде жазуда логикалық шартты тексеру үшін ромб қолданылады: ромбыдан екі бағыттың ("ия"- шарт орындалса, "жоқ"- шарт орындалмаса) бірімен жылжуға болады, себебі тармақталу барысында бір тармақ қана орындалады (3-сурет). Тармақтың саны есептің шартына байланысты 2-ден көп болуы да мүмкін (4-сурет).

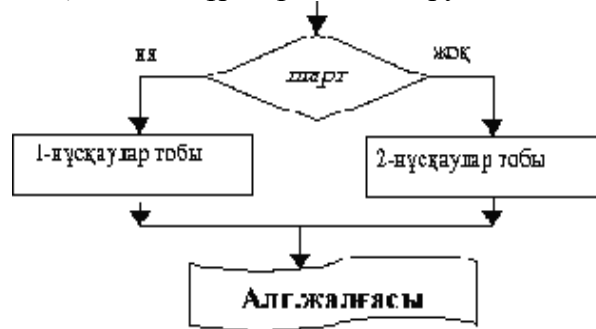
Логикалық немесе жәй ғана *шарт* деп екі өрнек арасындағы қатынасты айтады. Шартты -“ия”



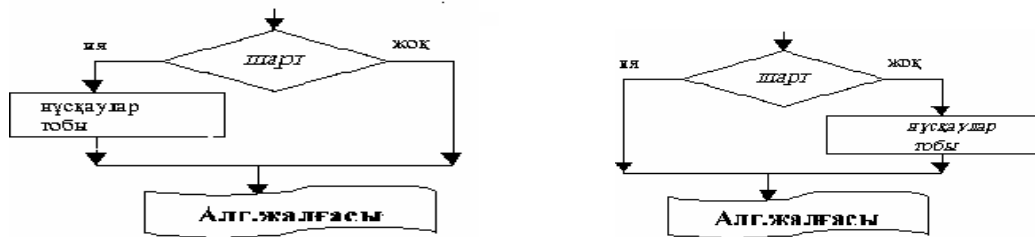
4-сурет

немесе “жоқ” деп жауап қайтаратын сұраулы сөйлем мағынасында түсінуге болады. Шарттың орындалмағаны - оған қарама-қарсы шарт орын алатындығын білдіреді, мысалы, $x > 0$ орындалмаса, $x \leq 0$ болғаны, $D=0$ шарты орындалмаса, $D < 0$ немесе $D > 0$ орындалғаны. Қарама-қарсы шарттардың біреуін тексерсе жеткілікті, екіншісі соның салдары болып табылады.

Тармақталуды *толық және ықшам* екі түрде ұйымдастыруға болады (5, 6-суреттер).



5-сурет. а), б) Тармақталудың толық түрі

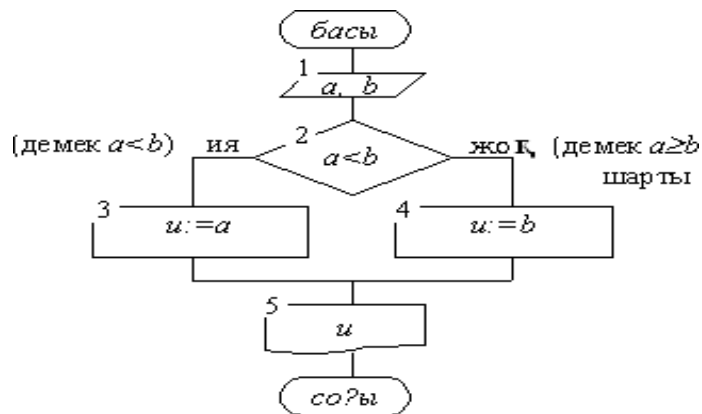


6-сурет. Тармақталудың ықшам түрі

2-мысал. Белгілі екі санның кішісін табу алгоритмін құру керек.

Шешуі: Есептің шартын $u = \min \{a, b\}$ түрінде жазуға болады, мұндағы u айнымалысы a немесе b ұяшықтарындағы мәндердің кішісі.

a мен b ұяшықтарына берілген мәндерін енгізген соң оларды өзара салыстырып, кішісіне a немесе b - u ұяшығына меншіктейтін алгоритмді суретте көрсетілген схема арқылы ұйымдастыруға болады. Бұл - тармақталған құрылымды алгоритм, мұнда тармақталудың толық схемасы қолданылған.

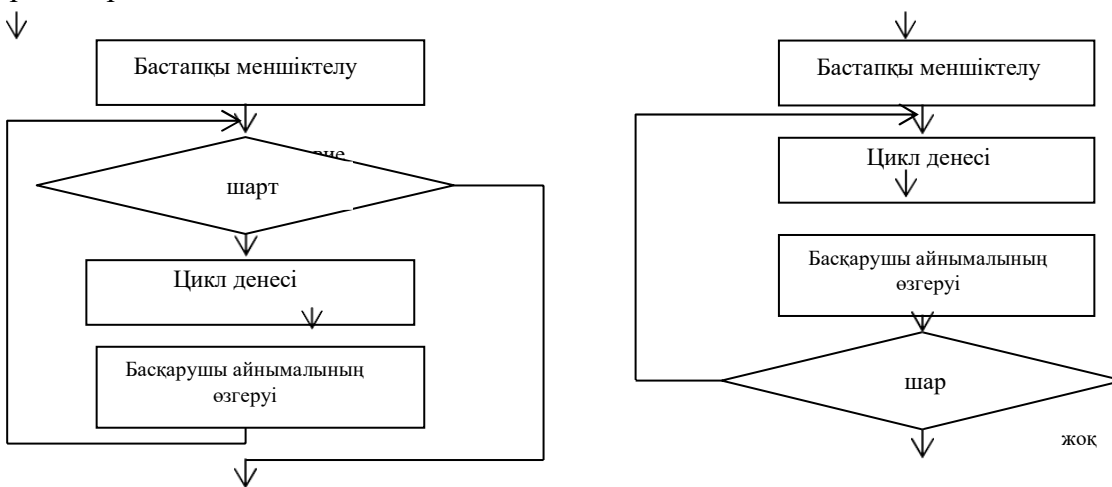


Циклдік алгоритм

Математикада, физикада, басқа да көптеген тәжірибелік есептерді шығару барысында бір математикалық формуланы құрайтын айнымалылардың әр-түрлі мәндерінде бірнеше рет қайталап есептеу жүргізу – жиі кездесетін жағдай. Осыған байланысты алгоритм құрғанда бір қатар әрекеттердің қайталанып орындалуын ұйымдастыру қажет болады. Алгоритмнің көп рет қайталап орындалатын бөліктерін *цикл* деп, ал қайталану әрекеттер тізбегін *цикл денесі* деп атайды. Циклдік алгоритмдерді пайдалану оларды кейіннен программаларда цикл операторы түрінде қысқартып жазуға мүмкіндік береді. Сонымен, бірнеше рет қайталанатын бөлігі бар алгоритмдер тобы *циклдік алгоритмдерге* жатады.

Алгоритмдегі әрекеттер өзінің жазылу ретіне сәйкес тізбектеліп немесе белгілі бір шартқа байланысты тармақталып немесе қайталанып орындалады. Яғни, алгоритмдегі әрекеттердің орындалу тәртібі белгілі бір нұсқаулар бойынша басқарылады. Осындай нұсқауларды басқару құрылымдары деп атайды.

Кез келген алгоритмді құруда басқару құрылымдардың маңызы зор болып табылады. Негізгі басқару құрылымдарына сызықтық және тармақталу құрылымдарымен қатар циклдік құрылымды алгоритмдері де жатады.



8. Сурет. қайталану (цикл) алгоритмдерінің негізгі типтері

8 суретте бейнеленген циклдік алгоритмдерінің ерекшеліктері мынада:

а) бұл қайталану құрылымында - цикл денесін құрайтын шығырлар мен цикл параметрінің өзгеруі циклдің қайталануын тексеретін шығырдан кейін орналасқан ("Дейін"-циклі);

б) бұл қайталану құрылымында - цикл денесін құрайтын шығырлар мен цикл параметрінің өзгеруі циклдің қайталануын тексеретін шығырға дейін орналасқан ("әзірше"-циклі).

Цикл қайталанған сайын өзінің мәнін өзгертетін айнымалыны *басқарушы айнымалы* немесе *циклдің параметрі* деп атаймыз. Циклдің параметрі – циклдің қайталану реттейтін, циклдің ішінде қайталану мөлшерін басқаратын айнымалы. Алгоритмнің орындалу барысында цикл параметрі – мысалы x , өзінің ең алғашқы x_0 мәнінен ең соңғы x_k мәніне дейін тұрақты h шамаға өзгеріп отырады. Осының нәтижесінде цикл параметрі төмендегі мәндерді қабылдайды:

$$x_0, x_0 + h, x_0 + 2h, x_0 + 3h, \dots, x_0 + (k-1)h, x_k$$

Циклдік қайталану саны келесі формула арқылы есептеледі:

$$n = \frac{x_k - x_0}{h} + 1$$

арқашанда бүтін сан болады, егер ол аралас сан болса, онда оның бөлшегі алынып тасталады, өйткені n қайталану санын анықтайтын шама оның мәні тек бүтін натуралдық сан болмақ.

Байқағанымыздай, белгілі бір мәліметтер жиынтығы мәндерінде сызықтық және тармақталған құбылыстар бір-ақ реттен орындалады, "Дейін" –циклі болса кем дегенде бір рет орындалады (өйткені цикл аяқталғаны жөніндегі шарттың бірінші тексерілуі цикл денесінің

әрекеттері орындалғаннан кейін болады), ал ”әзір” - циклі бір рет те орындалмауы мүмкін. Циклдерді қолдану алгоритм шығырларының көлемін және пограмманың ұзындығын қысқартып ықшамды түрде көрсетуге мүмкіндік береді.

Циклдер белгілі бір ережелер бойынша ұйымдастырылады. ”Дейін”-циклінің табиғи тілде жазылған алгоритмін қарастырайық.

1. Алдымен бастапқы мән меншіктеу жүргізіледі (цикл параметріне бастапқы мәні меншіктелінеді, қосынды жинақтау үшін бастапқы мән 0, ал көбейтінді жинақтау үшін бастапқы мән 1-ге тең деп алынады).

2. Циклдің денесінің әрекеттері орындалады (есептеу процесінің бірнеше рет қайталанатын бөліктері).

3. Әр қайталымға сәйкес циклдің параметрі өзгертіледі.

4. Цикл соңы немесе қайталану шарты тексеріледі.

5. Циклді басқару, яғни циклдің басына өту, егер циклдің қайталануы қажет болса, немесе циклдің қайталануы біткен соң одан шығу.

Циклдік алгоритмдерде қайталану саны алдына ала белгілі циклдер деп және цикл денесінің орындалу саны алдын ала белгісіз деп бөлуге болады. Бірінші жағдайда оларды *реттелген циклдер* деп ал екінші жағдайда, *итерациялық циклдер* деп ажыратады.

Қайталану саны алдына ала белгілі циклдерде цикл параметрінің бастапқы және соңғы мәндер белгілі болады. Цикл параметрінің мәні белгілі бір айқын заңдылық бойынша, цикл қайталанған сайын өзгеріп отырады. Осы заңдылық арқылы циклдің қайталану саны анықталады. Итерациялық циклдерде өзгеру заңы ашық түрде берілмейді.

Бақылау сұрақтары

1. Циклдік құрылымды алгоритмдерінің ерекшеліктері неде?
2. Циклдің параметрі деген не?
3. Циклдік алгоритмдерінің іс-әрекеттері қандай кезектілікпен орындалады (циклдарды ұйымдастыру ережелері)?
4. Қосындыны және көбейтіндіні жинақтау есептерінде циклді ұйымдастырудағы әрекеттердің кезектілігін және олардың ерекшелігін атаңыз.
5. Функция мәнін кестелеу есептерінде цикл ұйымдастырудағы әрекеттер кезектілігін және олардың ерекшелігін атаңыз.
6. Модификатор шығырының ішінде қандай информация жарияланады?
7. Модификатор шығырының кескін үйлесімі және орындайтын әрекетін түсіндіріңіздер.
8. ”Дейін” және ”әзір” циклдік алгоритмдерінің орындалу ерекшелігін көрсетіңіздер.
9. Типтік циклдік құрылымды алгоритмдерді атаңыз.
10. Цикл параметрінің қадамы дегеніміз не, және ол қандай мәндерді қабылдай алады?

Әдебиеттер тізімі:

1. С.А.Немнюгин. Программирование на языке высокого уровня. Turbo Pascal: Учебник. 2-е изд.-СПб.: Издательство «Питер», 2003.
2. С.А.Немнюгин. Turbo. Pascal. Практикум. 2-е изд.-СПб.: Издательство «Питер», 2005.
3. Гусева А.И. Учимся программированию. PASCAL. 7. Задачи и методы их решения. - 2-е изд, перераб, и доп.-М.; “ Диалог -МИФИ“, 2003