

Veprimet e përsëritura. Algoritmet ciklike

Për ato probleme, zgjidhja e të cilave kërkon përsëritjen disa herë të një veprimi apo të një grupi veprimesh, nuk mjaftojnë strukturat algoritmike që kemi parë deri tani. Për ndërtimin e algoritmeve që përmbajnë përsëritje veprimesh, dhe për këtë arsye quhen edhe algoritme ciklike, duhen përdorur veprimet e përsëritura ose ciklet të cilat do t'i përshkruajmë më poshtë.

1 Cikli “Derisa - kryej”

Forma e këtij veprimi është:

Derisa (kusht logjik) kryej

Fillim

veprim;

veprim;

.....

Veprim;

Fund;

**blok veprimesh ose
një veprim i vetëm.**

Në qoftë se vlera e kushtit logjik është e gënjeshtërt, atëherë nuk kryhet blloku i veprimeve që përfshihet ndërmjet fjalës kryej dhe pikëpresjes së fundit por vazhdohet me veprimin pasardhës.

Në qoftë se vlera e kushtit është e vërtetë, atëherë kryhet blloku i veprimeve dhe rivlerësohet përsëri kushti logjik. Në qoftë se ai ka përsëri vlerën e vërtetë, kryhet përsëri blloku i veprimeve, e kështu me radhë, derisa kushti të marrë vlerën e gënjeshtërt.

Vërejtje: Për të siguruar ndalimin e ciklit (e përsëritjes së veprimeve) duhet që të paktën njëri nga veprimet e bllokut të veprojë mbi madhësitë që marrin pjesë në kushtin logjik, në mënyrë që të vijë një moment që vlera e kushtit të bëhet e gënjeshtërt, dhe për pasojë të dilet nga cikli. Në rast të kundërt, kemi të bëjmë me “cikël pa fund” ose thuhet që algoritmi “rrotullon”.

Skematikisht cikli **Derisa** paraqitet në figurën 1.

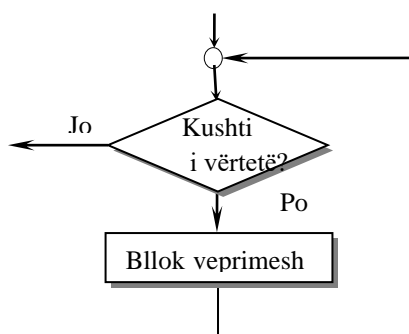


Figura 1. Algoritëm ciklik. Struktura **Derisa**.

Për të ilustruar përdorimin e këtij veprimi le të shohim shembullin e mëposhtëm:

Shembull 1: Të llogaritet shuma:

$$S = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/k + \dots$$

ku secili term i shumës duhet të plotësojë kushtin :

$$1/k \geq \text{Eps} \quad (\text{Eps} \geq 0 \text{ është një numër i dhënë})$$

Shumën do ta llogarisim duke i mbledhur të gjithë termat njëri pas tjetrit në variabëlën që po e quajmë S. Në fillim do t'i japim këtij variabëli vlerën zero. Pastaj i shtojmë atij numrin 1, pastaj numrin 1/2, 1/3, e kështu me radhë duke kontrolluar në çdo hap nëse termi që do t'i shtojmë variabëlës S është bërë më i vogël se Eps. Në rast se ai bëhet më i vogël se Eps, atëherë duhet të ndalemi, sepse në S kemi shumën e kërkuar.

Po emërtojmë dy variabëla të tjerë T dhe Em përkatësisht për termin e shumës dhe për emëruesin e termit.

Është e qartë që variabëli Em do të marrë vlerat 1, 2, 3,...., pra për të kaluar nga një term në termin pasardhës mjafton të kryhet veprimi i vlerë dhënies si më poshtë:

$$Em = Em + 1;$$

dhe kjo vlerë e re e emëruesit t'i jepet termit:

$$T = 1 / Em;$$

Futja e termit të ri në shumë do të realizohet me anë të veprimit:

$$S = S + T;$$

Pra algoritmi ynë do të përmbajë këto veprime:

Së pari do të lexohet vlera e dhënë Eps. Më pas variablave S, Em, T do t'u jepen vlerat fillestare përkatësisht 0, 1, 1, pasi emëruesi i termit të parë është 1 dhe vetë ky term e ka vlerën po 1. Ndërkaq, në shumë nuk është shtuar akoma asnjë term, kështu që vlera në variabëlin S do të jetë 0.

Tre veprimet që do të pasojnë: futja e termit në shumë dhe përftimi i termit pasardhës (me anë të veprimeve: $E = Em + 1$; $T = 1/Em$;) do të përsëriten sa kohë që termi i shumës do të jetë më i madh se Eps.

Për të realizuar këtë përsëritje do të përdorim veprimin **Derisa – kryej**, për të cilin po diskutojmë.

Më poshtë, jepet algoritmi i shprehur në pseudo kod për llogaritjen shumës së mësipërme:

(*Algoritëm Shuma 1*)

Fillim

lexo(Eps) ;

S = 0 ;

Em = 1 ;

T = 1 ;

Derisa T > = Eps **kryej**

Fillim

S = S + T ;

Em = Em + 1 ;

T = 1/Em ;

Fund ;

afisho (S) ;

Fund.

2 Cikli “Përsërit - gjersa”

Forma e këtij veprimi është :

Përsërit

veprim ;

veprim ;

.....

veprim ;

Gjersa < kusht logjik >

Në **ndryshim** nga cikli Derisa - kryej, blloku i veprimeve kryhet sa kohë që vlera e kushtit logjik është **jo**. Sapo kushti logjik të marrë vlerën **po**, blloku i veprimeve nuk kryhet më dhe vazhdohet me veprimin pasardhës.

Skematikisht cikli Përsërit paraqitet në figurën 2.

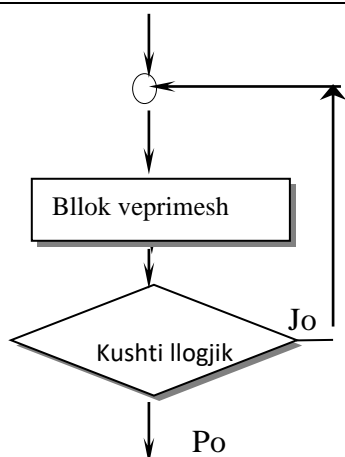


Figura 2 Algoritëm ciklik Struktura Bëj - Gjersa.

```

Algoritëm shuma2;
Fillim
    lexo(Eps);
    S = 0;
    Em = 1;
    T = 1;
Përsërit
    Fillim
        S = S + T;
        Em = Em + 1;
        T = 1/Em;
    Fund;
gjersa T < Eps;
afisho (S);
  
```

Fund.

Në këtë strukturë blloku i veprimeve kryhet të paktën një herë gjersa të bëhet kontrolli.

3 Cikli “Për - deri - kryej”

Në qoftë se numri i përsëritjeve është i njohur që në fillim, është më e përshtatshme të përdoret struktura ciklike Për - deri – kryej. Forma e këtij veprimi është:

Për var.kontrolli = *vl.fill* deri *vl.përf* me hap *hp* Veprim(e);

Kjo strukturë lejon të kryhet në mënyrë të përsëritur një instruksion apo një bllok instruksionesh, ndërkohë që **variabëliti të kontrollit ciklik** i jepen **vlera progresive**. Pra, veprimi apo blloku i veprimeve do të kryhen për vlerat e variabëliti të kontrollit nga vlera fillestare (*vl.fill*), deri në vlerën përfundimtare (*vl.përf*) duke ndryshuar në çdo cikël me hapin *hp*.

Kur vlera e këtij variabëli e kapërcen vlerën përfundimtare, blloku nuk përsëritet më, por kalohet në veprimin pasardhës. Skematikisht ky veprim paraqitet në figurën 3.

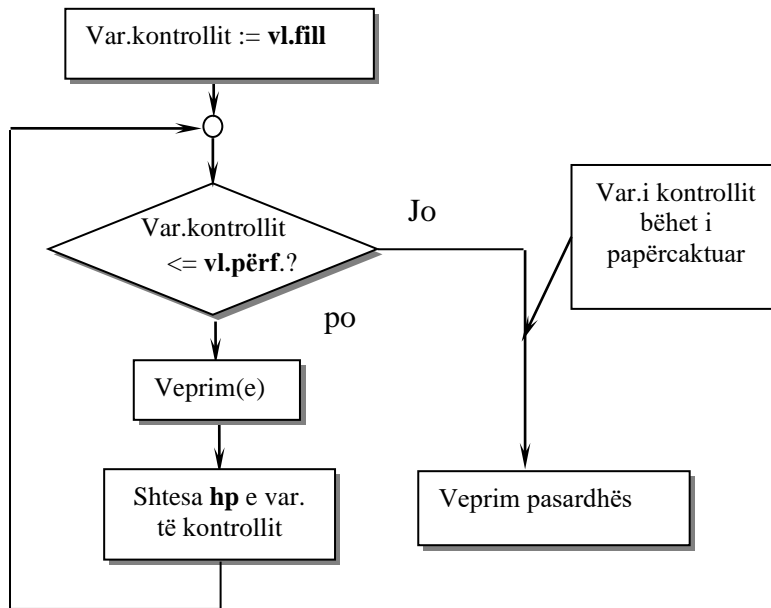


Figura 3. Algoritëm ciklik. Struktura për - deri - kryej.

Në bllok skemën e mësipërme është paraqitur rasti kur vlera fillestare e variabël të kontrollit ciklik është më e vogël se vlera përfundimtare. Le të aplikojmë këtë strukturë në shembullin e mëposhtëm :

Shembull 5. Të llogaritet shuma:

$$S=1 + 1/2 + 1/3 + \dots + 1/100$$

Le të shënojmë me T termin e shumës dhe me Em një variabël, i cili do të marrë për vlerë emëruesit e termave, pra Em = 1, 2, 3,, 100.

Atëherë veprimet:

$$T= 1/Em;$$

$$S = S + T;$$

do të përsëriten 100 herë, për vlera të variabël Em nga 1 deri në 100 duke u shtuar me nga 1 në çdo përsëritje të ciklit. Është e qartë që në këtë rast mund të përdorim ciklin Për - deri – kryej, siç tregohet në algoritmin e mëposhtëm:

(* Algoritëm shuma3 *)

Fillim

$$S = 0;$$

për Em =1 deri 100 kryej

Fillim

$$T = 1 / Em;$$

$$S = S + T;$$

Fund;

afisho (S);

Fund.

Shihet që vlera e emëruesit **Em shërben** në këtë shembull **dhe si variabël i kontrollit** për ciklin. Siç kemi theksuar edhe në fillim të këtij kapitulli, për zgjidhjen e një problemi me ndihmën e kompjuterit, nuk mjafton vetëm ndërtimi i algoritmit, por është e domosdoshme që ky i fundit të shprehet në një gjuhë programimi.