

Mavzu: I) Oddiy tsiql algoritmlarini dasturlash.

Kurs ishini bajarishdan maqsad: Pascal tilida takrorlanuvchi jarayonni xosil qilish:

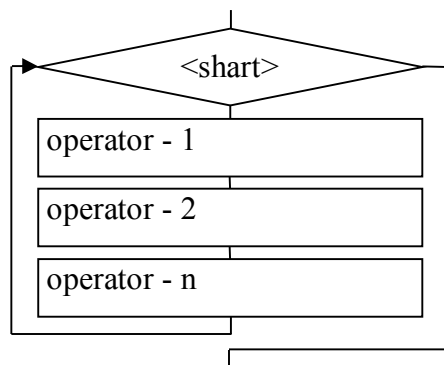
- A) «toki» – oldindan qo'yilgan shart bilan takrorlanuvchi jarayonni tashkil qilish;
- B) «gacha» – oxirida qo'yilgan shart bilan takrorlanuvchi jarayonni tashkil qilish;
- C) Parametrli tsiql yordamida takrorlanuvchi jarayonni xosil qilishni o'rganish va tsiklga doir dasturlar tuzish.

Nazariy qism:

Malum bir amallar ketma-ketligining o'zgaruvchilarning turli qiymatlarida bir necha bor qayta – qayta takrorlanishi *takrorlanuvchi jarayon* yoki *tsiql* deyiladi. Takrorlanuvchi jarayon ichiga kiruvchi amallar *takrorlanish sohasi* deb yuritiladi. Bu jarayonda qiymatini o'zgartirib turuvchi o'zgaruvchi *tsikl parametri* deb ataladi. Takrorlanuvchi jarayonni uch xil ko'rinishda tashkil kilish mumkin.

«toki» – oldindan qo'yilgan shart bilan takrorlanuvchi jarayonni tashkil qilish

Oldindan qo'yilgan shart bilan – «toki». Bu struktura quyidagi rasmdagi ko'rinishga ega. Bu strukturada takrorlanuvchi jarayon mantiqiy ifodaning qiymatiga qarab bajaradi yoki bajarmaydi. Agar mantiqiy ifodaning qiymati **true** bo'lsa, tsiql bajarilaveradi, aks xolda tsiql bajrilmaydi. Bu strukturada tsiql bir marta ham bajarilmasligi mumkin.



Pascal tilida bu strukturani quyidagich yozishimiz mumkin.

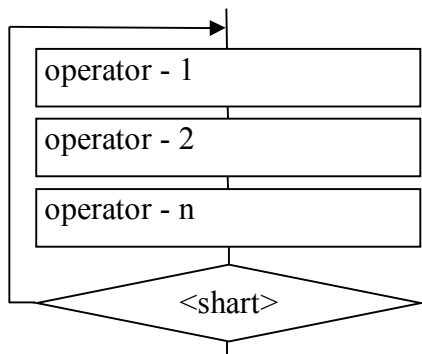
```
while <shart> do  
begin  
    <operator – 1>;  
    <operator – 2>;  
    <operator – n>;  
end;
```

Bu erda **while** – «toki» do – bajarish xizmatchi so'zlar, <shart> – mantiqiy ifoda yoki mantiqiy shart, tsiql soxasining operatorlari – tsiqlga kiruvchi

operatorlar ketma-ketligi. Agar bu ketma-ketlik bitta operatoridan iborat bo'lsa, **begin** va **end** ni yozish shart emas.

«gacha» – oxirida qo'yilgan shart bilan takrorlanuvchi jarayonni tashkil qilish

Bu strukturada mantiqiy ifodaning qiymati **false** bo'lsa, qayta bajariladi, aks xolda **true** da tsikldan chiqib ketiladi. Bu strukturada tsikl soxasi xech bo'lmaganda bir marta bajariladi.



Pascal tilida bu strukturani quyidagich yozishimiz mumkin.

repeat

<operator – 1>;

<operator – 2>;

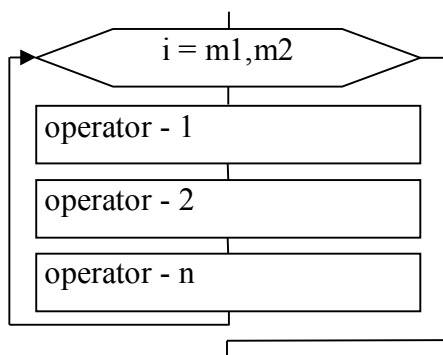
<operator – n>;

until <shart>;

Bu erda **repeat** va **until** kalit so'zlari. <shart> o'rnida mantiqiy ifoda qo'llaniladi.

Parametrli tsicl yordamida takrorlanuvchi jarayonni xosil qilishni

Parametrli tsicl tashkil qilish. Bu struktura quyidagi ko'rinishga ega



Bu erda **i** – tsikl parametri bo'lib, u jarayonning takrorlanishlar sonini belgilaydi. **m1** – tsikl parametrining boshlang'ich qiymati

m2 – tsikl parametrining oxirgi qiymati

Pascal tilida bu strukturani quyidagich yozishimiz mumkin

for i:=m1 to m2 do

begin

<operator – 1>

<operator – 2>

<operator – n>

end;

Pascal tilida for tsikli faqat 1 qadam bilan o'zgaradi. Agar m1 ni qiymati m2 dan katta bo'lsa uni kamayish bo'yicha ham -1 qadam bilan tsicl tashkil qilish mumkin. U quyidagicha yoziladi

for i:=m1 downto m2 do

begin

<operator – 1>

<operator – 2>

<operator – n>

end;

Izoh: yendi hurmatli talabalar takrorlanuvchi jarayonlarni ko'rsatish uchun bitta misol olamiz va u misolni uchta usul bilan echamiz.

Misol

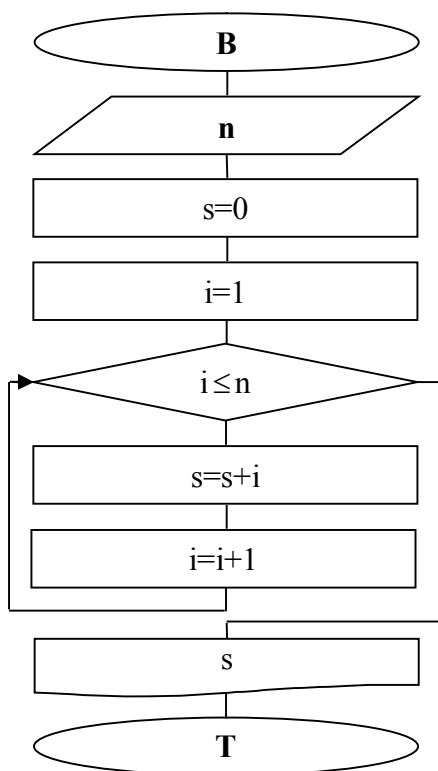
Misol1: 1 dan n gacha bo'lgan sonlarni yig'indisini toping.

Matematik rasmiylashtirish:

1. Matematik model:

$$s = \sum_{i=1}^n i$$

2. Formulaga mos algoritm quramiz:



3. Algoritmga mos dastur tuzamiz.

```

program while;
var
    n,s,i:integer;
begin
    readln(n);
    s:=0;
    i:=1;
    while (i<=n) do
        begin
            s:=s+i;
            i:=i+1;
        end;
    writeln(s);
end.
  
```

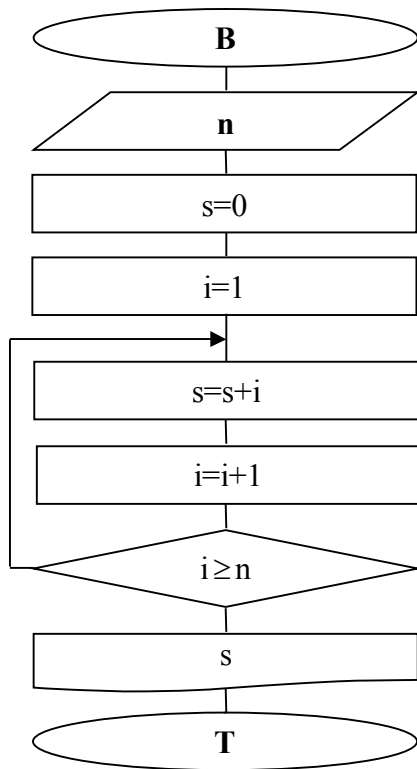
Misol2: 1 dan n gacha bo'lgan sonlarni yig'indisini toping.

Matematik rasmiylashtirish:

1. Matematik model:

$$s = \sum_{i=1}^n i$$

2. Formulaga mos algoritm quramiz:



3. Algoritmga mos dastur tuzamiz.

```
program while;
var
    n,s,i:integer;
begin
    readln(n);
    s:=0;
    i:=1;
    repeat
        s:=s+i;
        i:=i+1;
    until (i>=n);
    writeln(s);
end.
```

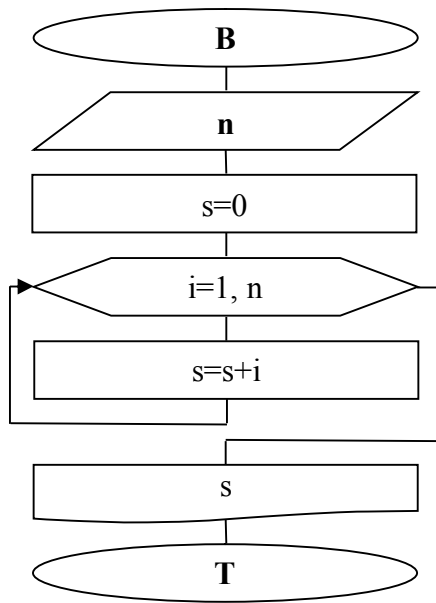
Misol3 : 1 dan n gacha bo`lgan sonlarni yig`indisini toping.

Matematik rasmiylashtirish:

1. Matematik model:

$$s = \sum_{i=1}^n i$$

2. Formulaga mos algoritm quramiz:



3. Algoritmga mos dastur tuzamiz.

```
program for;  
var  
    n,s,i:integer;  
begin  
    readln(n);  
    s:=0;  
    for i:=1 to n do  
        begin  
            s:=s+i;  
        end;  
    writeln(s);  
end.
```

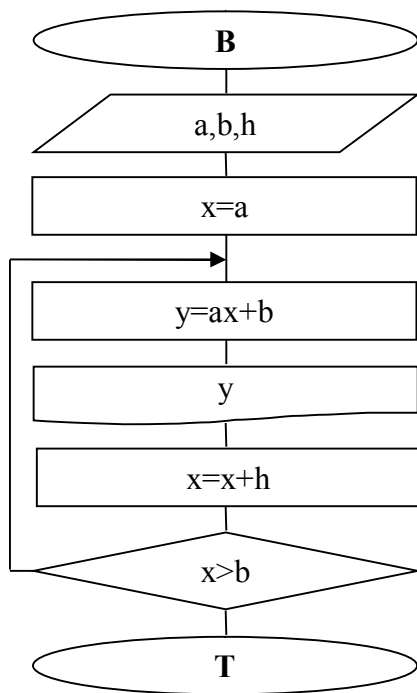
Misol4: $y = ax + b$, $a \leq x \leq b$, h - berilgan haqiqiy son.

Matematik rasmiylashtirish:

1. Matematik model:

$$y = ax + b, \quad a \leq x \leq b, \quad h$$

2. Formulaga mos algoritm quramiz:



3. Algoritmga mos dastur tuzamiz.

program example;

var

h,x,y:real;

a,b:integer;

begin

readln(a,b,h);

x:=a;

repeat

y:=a*x+b;

writeln(y);

x:=x+h;

until (x>b);

end.

Mavzu: II) Murakkab tsiql algoritmlarini dasturlash.

Kurs ishini bajarishdan maqsad: Pascal tilida murakkab takrorlanuvchi jarayonni xosil qilish: Biz oldingi laboratoriyada o`rgangan FOR, WHILE va REAPEA...UNTIL tsiqlarini murakkab jarayonlarga qo`llashni o`rganishdir.

Nazariy qism:

Malum bir amallar ketma-ketligining o`zgaruvchilarning turli qiymatlarida bir necha bor qayta – qayta takrorlanishi *takrorlanuvchi jarayon yoki tsiql* deyiladi. Takrorlanuvchi jarayon ichiga kiruvchi amallar *takrorlanish sohasi* deb yuritiladi. Bu jarayonda qiymatini o`zgartirib turuvchi o`zgaruvchi *tsikl parametri* deb ataladi. Odatda murakkab jarayonlar uchun tsiqlar ishlatiladi. Biz oldingi laboratoriyada siqlarni turlarini va ularni ishlash prinsipini o`rganib chiqdik va sodda dasturlarni echishda qo`lladik. Yendi murakkab jarayonlar uchun shu tsiqlarni qo`llab misollar yechamiz. Bu laboratoriyada asosan uchta tipdagi murakkab algoritmlarni ishlashni o`rganamiz. Har bir tur uchun 1 ta misol ko`rib uni sharhlaymiz.

Misol

Misol1: Umumiy yigindini hisoblash.

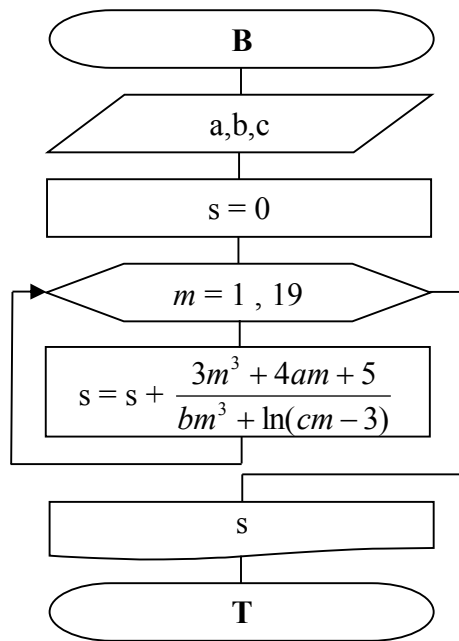
$$S = \sum_{m=1}^{19} \frac{3m^3 + 4am + 5}{bm^3 + \ln(cm - 3)}$$

Matematik rasmiylashtirish:

1. Matematik model:

$$S = \sum_{m=1}^{19} \frac{3m^3 + 4am + 5}{bm^3 + \ln(cm - 3)}$$

2. Formulaga mos algoritm quramiz:



3. Algoritmga mos dastur tuzamiz.

```

program summa;
var
  m:integer;
  a,b,c,s:real;
begin
  read(a,b,c);
  s:=0;
  for m:=1 to 19 do
    s:=s+(3*m*m*m+4*a*m+5)/(b*m*m*m+ln(c*m-3));
  writeln(s);
end.
  
```

Misol2: Umumiy ko'paytmani hisoblash.

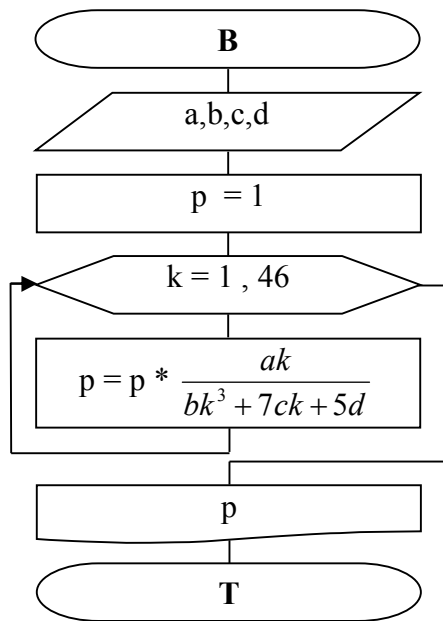
$$P = \prod_{k=1}^{46} \frac{ak}{bk^3 + 7ck + 5d}.$$

Matematik rasmiylashtirish:

1. Matematik model:

$$P = \prod_{k=1}^{46} \frac{ak}{bk^3 + 7ck + 5d}.$$

2. Formulaga mos algoritm quramiz:



3. Algoritmga mos dastur tuzamiz.

```

program multiplay;
var
  a,b,c,d,p:real;
  k:integer;
begin
  read(a,b,c,d);
  p:=1;
  for k:=1 to 46 do
    p:=p*(a*k/(d*k*k*k+7*c*k+5*d));
    writeln(a,b,c,d,p);
  end.
  
```

Misol3: Ko'paytmalarning umumiy yig'indisini hisoblash.

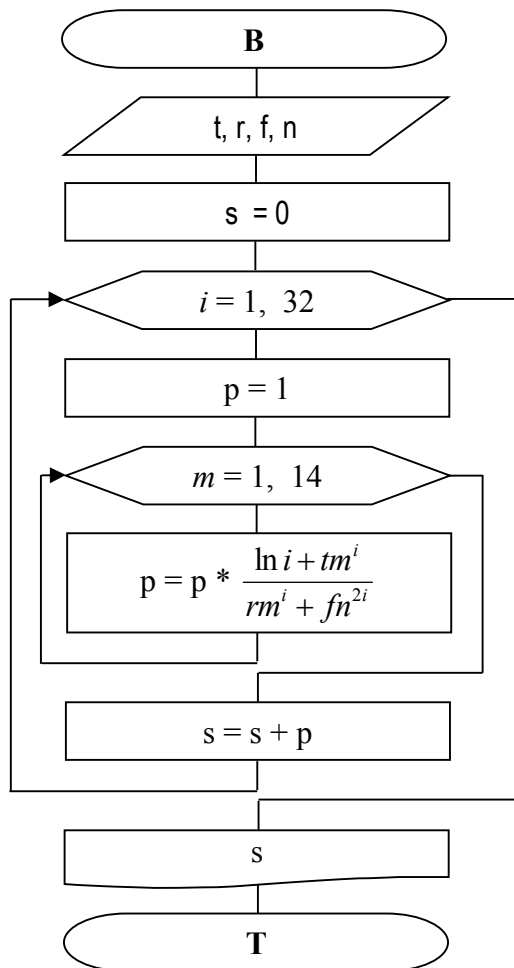
$$S = \sum_{i=1}^{32} \prod_{m=1}^{14} \frac{\ln i + tm^i}{rm^i + fn^{2i}}.$$

Matematik rasmiylashtirish:

1. Matematik model:

$$S = \sum_{i=1}^{32} \prod_{m=1}^{14} \frac{\ln i + tm^i}{rm^i + fn^{2i}}$$

2. Formulaga mos algoritm quramiz:



3. Algoritmgaga mos dastur tuzamiz.

```

program summaAndMultiplay;
var
  t,r,f,s,p:real;
  i,m,n: integer;
begin
  read(t,r,f,n);
  s:=0;
  for i:=1 to 32 do
    begin
      p:= 1;
      for m :=1 to 14 do
        p:=p*(ln(i)+t*exp(i*ln(m))/(r*exp(i*ln(m))+f*exp(2*i*ln(n))));
        s:=s + p;
      end;
    writeln(s);
  end.
  
```