

**Botirov D.B.**

*O'zbekiston, JDPU dotsent*

**Ulug'murodova L.D.**

*O'zbekiston, JJST katta o'qituvchi.*

**Komilova Z.X.**

*O'zbekiston, FarDU o'qituvchi*

**DASTURLASH TILLARI FANIDAN ICHMA–ICH JOYLASHGAN TAKRORLANISH  
JARAYONLARINI HISOBLASH BO‘YICHA AMALIY MASALLARNI HAL QILISHDA  
QIYOSLASH**

**СРАВНЕНИЕ РЕШЕНИЯ ПРАКТИЧЕСКИХ ЗАДАЧ ПО РАСЧЕТУ  
ВНУТРЕННЕ РАСПОЛОЖЕННЫХ ПОВТОРЯЮЩИХСЯ ПРОЦЕССОВ  
ИЗ НАУКИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ**

**COMPARISON OF SOLUTION OF PRACTICAL PROBLEMS ON  
CALCULATION OF INTERNALLY LOCATED REPETITIVE PROCESSES  
FROM THE SCIENCE OF PROGRAMMING LANGUAGES**

***Annotatsiya:** Ushbu maqolada ichma–ich joylashgan takrorlanish jarayonlarini hisoblash bo'yicha amaliy masallarni hal qilishdagi qiyoslash masalasi qaralgan bo'lib dastur kodini yozishda bugungi kunda ommalashib borayotgan Python dasturlash tilidan foydalanilgan.*

***Kalit so'zlar:** Algoritm, ichma-ich takrorlanuvchi algoritm, blok-sxema, jadval, for operatori, while operatori, ma'lumotlar tahlili, Python.*

***Аннотация:** В данной статье рассматривается задача сравнения при решении практических задач по расчету вложенных итерационных процессов, а для написания программного кода используется популярный сегодня язык программирования Python.*

***Ключевые слова:** алгоритм, вложенный рекурсивный алгоритм, блок-схема, таблица, оператор for, оператор while, анализ данных, Python.*

***Abstract:** In this article, the problem of comparison in solving practical problems on the calculation of nested iteration processes is considered, and the Python programming language, which is popular today, is used to write the program code.*

***Keywords:** Algorithm, nested recursive algorithm, block diagram, table, for operator, while operator, data analysis, Python.*

Hozirgi jadal rivojlanish va turli jarayonlarni avtomatlashtirish hamda robotlashtirish davrida dasturlashni bilish va uni o'z ish jarayonida ishlata olish texnik va pedagogik yo'nalishda ta'lim olayotgan o'quvchi-talabalar uchun juda muhim deb hisoblanadi. Bu zamonaviy mutaxassislar uchun eng zaruriy talablardan biridir. Sababi hozirgi kunda informatika turli-tuman sohalarda muvaffaqiyatli ravishda qo'llanilishi mumkinligini hech kim ham rad eta olmaydi[1].

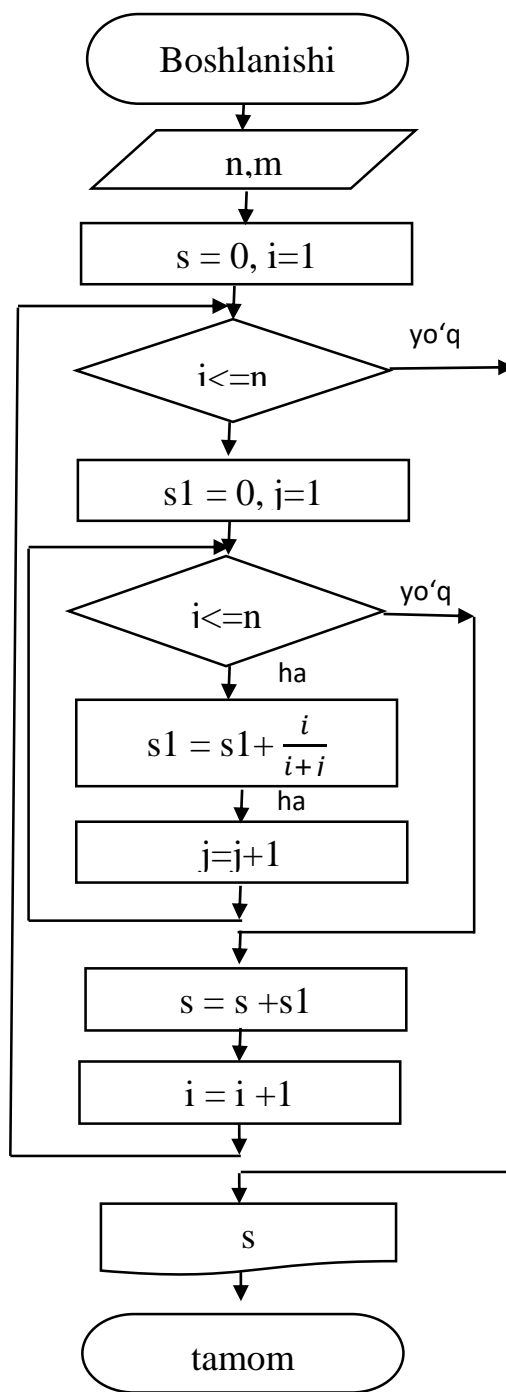
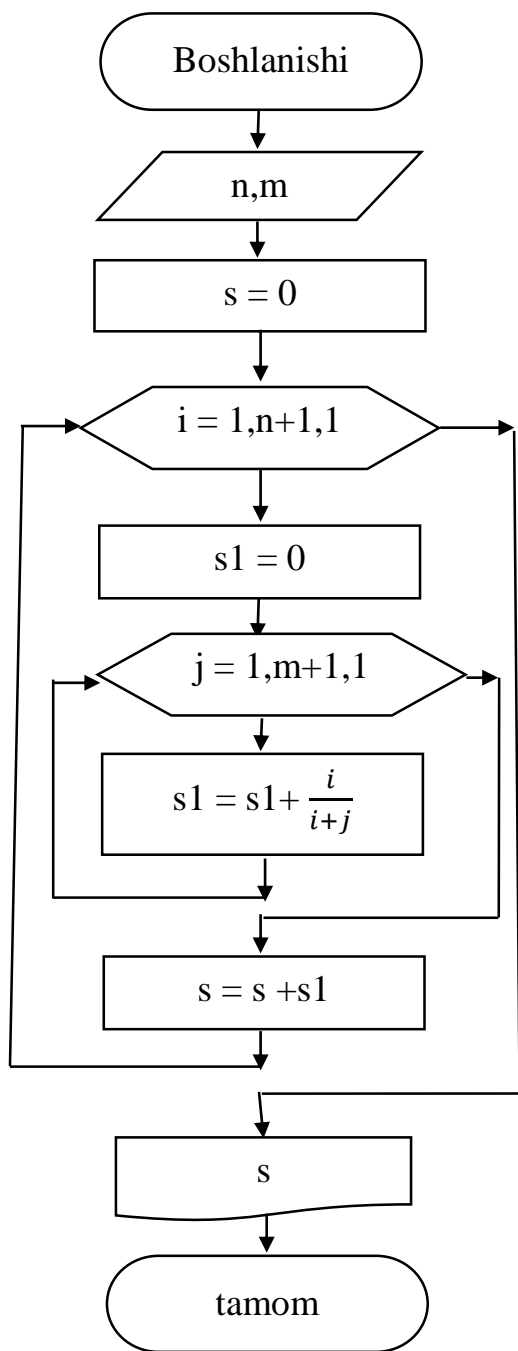
Ichma-ich algoritm deganda, biror maqsadga erishishga qaratilgan ijrochi bajarishi uchun mo'ljallangan ko'rsatma (buyruq)larning aniq, tushunarli va chekli ketma-ketligi tushuniladi. Algoritm ijrochisi- algoritmda ko'rsatilgan buyruq yoki

ko'rsatmalarni bajara oladigan abstrakt yoki real (texnik yoki biologik) sistema. Takrorlanuvchi algoritmlar deb, biron bir shart tekshirilishi yoki qandaydir parametrning har xil qiymatlari asosida algoritmda takrorlanish yuz beradigan jarayonlarga aytiladi[2].

Shunday jarayonlar ham bor, ularning ayrim bo'laklari bir necha marta takrorlanadi. Oddiy takrorlash algoritmlarida bir parametrning ma'lum qiymatida takror hisoblashlar bajariladi. Ichma-ich joylashgan takrorlashlarda esa birinchi parametrning birgina qiymati uchun ikkinchi parametrning barcha qiymatlarida hisoblash jarayoni bajariladi. Demak birinchi parameter bilan hosil qilingan takrorlash tashqi ikkinchi parameter bilan ichki takrorlanish deb ataladi. Biz quyida ayrim misollarni qarab chiqamiz. Quyidagi yig'indini hisoblash talab qilinsin. Bu yig'inda qanday qo'shiluvchilardan iborat ekanligini yozib chiqamiz, yani

$$\sum_{i=1}^n \sum_{j=1}^m \frac{i}{i+j} = \frac{1}{1+1} + \frac{1}{1+2} + \dots + \frac{1}{1+m} + \frac{2}{2+1} + \frac{2}{2+2} + \dots + \frac{2}{2+m} + \dots + \frac{n}{n+1} + \frac{n}{n+2} + \dots + \frac{n}{n+m}$$

shaklida bo'ladi. Ko'rinib turibdiki i parametrning bitta qiymatiga j parametr to'liqlikicha bajariladi, shu jarayonni hisoblash algoritmini blok-sxemasini tuzamiz.



Bunday jarayonlar uchun algoritmlar tuzishda takrorlanuvchi algoritmlardan foydalaniladi. Odatda algoritmlarni tasvirlashning blok-sxema usuli keng tarqalgan[3].

Takrorlashlar soni ma'lum bo'lgan va takrorlanish soni noma'lum bo'lgan jarayonlar uchun alohida alohida blok-sxema tuzdik, endi bu blok-sxema yordamida dasturning kodini yozamiz.

#ichma-ich joylashgan takrorlash

#ichma-ich joylashgan

```

#for operatori orqali
from math import*
n=int(input(" i uchun hadlar soni "))
“))
m=int(input("j uchun hadlar soni "))
“))
s=0
for i in range(1,n+1,1):
    s1=0
    for j in range(1,m+1,1):
        s1= s1+i/(i+j)
    s=s+s1
    print("s=",s)

```

```

#while operatori orqali
from math import*
n=int(input(" i uchun hadlar soni
m=int(input("j uchun hadlar soni
s=0
i=1
while (i<=n):
    s1 = 0
    j=1
    while (j<=m):
        s1=s1+i/(i+j)
        j=j+1
    s=s+s1
    i=i+1
    print("s=",s)

```

Ikkala holatda ham takrorlanishlar soni  $n \cdot m$  marta bajariladi, Takrorlanuvchi algoritmlar “ $i=i+1$ ,” $j=j+1$ ” “ $s1=s1+i/(i+j)$ ” yoki “ $s=s+s1$ ” ko‘rinishidagi ko‘rsatmalarning ishtiroki bilan ajralib turadi. Bunday ko‘rsatmalarning mohiyatini tushunish uchun takrorlanishning bir nechta qadamini ko‘rib chiqish lozim.

Shart oldin tekshiriluvchi (toki) takrorlanuvchi algoritmlarda avval shart tekshiriladi, so‘ngra, agar shart qanoatlantirsa (rost bo‘lsa), sikl tanasi bajariladi, aks holda hisoblash to‘xtatiladi.

Shart keyin tekshiriluvchi (gacha) takrorlanuvchi algoritmda avval sikl tanasi bajarilib, so‘ngra sikldan chiqish sharti tekshiriladi, ya’ni sikl tanasi qo‘yilgan shart bajarilib bo‘lguncha takrorlanaveradi. Takrorlanish jarayonini tahlil qilish uchun, i uchun 1 dan n gacha bo‘lgan tashqi siklni va j uchun 1 dan m gacha bo‘lgan ichki siklni hisoblash algoritmini ko‘rib chiqamiz.

Bu algoritmni shart oldin tekshiriluvchi algoritm sifatida tuzib chiqamiz.

Tuzilgan blok-sxema asosida  $n=4$  va  $m=4$  holati uchun i, j, s1 va s larning takrorlanish jarayonidagi har bir qiymatini hisoblab quyidagi jadvalni hosil qilamiz: jarayonidagi har bir qiymatini hisoblab quyidagi jadvalni hosil qilamiz:

Qadam	i–tashqi sikl	j –ichki sikl	s1–ichki yig‘indi	s-tashqi yigindi
1-qadam	$i=1$	$j=1$	$s1=s1+1/(1+1) = \frac{1}{2}$	$s= 0 + \frac{1}{2} = \frac{1}{2}$

2-qadam	$i=1$	$j=j+1=1+1=2$	$s1=s1+1/(1+2) = \frac{5}{6}$	$s=\frac{1}{2} + \frac{5}{6} = \frac{8}{6}$
3-qadam	$i=1$	$j=j+1=2+1=3$	$s1=s1+1/(1+3) = \frac{13}{12}$	$s=\frac{8}{6} + \frac{13}{12} = \frac{29}{12}$
4-qadam	$i=1$	$j=j+1=3+1=4$	$s1=s1+1/(1+4) = \frac{77}{60}$	$s=\frac{29}{12} + \frac{77}{60} = \frac{111}{30}$

Ushbu algoritm asosida python dasturlash tilida tuzilgan dastur yordamida va jadval bo'yicha qiymatlarni solishtiramiz.

Bu yerda, ikkala dasturdan olingan natijalar aynan jadvaldagi qiymatlar bilan bir xil. Ichma-ich takrorlanuvchi algoritmlarda har bir olingan natija bilan tanishish ushbu algoritmlarni yanada chuqurroq o'zlashtirishga yordam beradi.

#### Foydalanilgan adabiyotlar:

1. Sh.A.Mengliyev, O.A.Abdug'aniyev, S.Q.Shonazarov, D. Sh.To'rayev. Python dasturlash tili. O'quv qo'llanma. Termiz, 2021.
2. A.R.Azamatov. Algoritmash va dasturlash asoslari. O'quv qo'llanma. Toshkent, 2013.
3. S.Eshtemirov, F.M.Nazarov. Algoritmash va dasturlash asoslari. O'quv qo'llanma. Samarqand, 2018.
4. A.Sattarov, B.Qurmanboyev Informatika va hisoblash texnikasi asoslari. Toshkent "O'qituvchi" 1996, 224 bet.
- 5.С.А.Абрамов и др. Задачи по программированию. М., «НАУКА» 1988 г.