



DASTURLASHDA ALGORITMIK YONDASHUVLAR: USLUBLAR, STRUKTURAVIY FARQLAR VA ANALITIK KO'RINISH

Shamsitdinova Gavxaroy Azmiddin qizi

NamDU Fizika-matematika fakulteti talabasi

Shamsitdinovagavharoy2005@gmail.com

Ilmiy raxbar: Dadaxanov Musoxon Xoshimxonovich

NamDU "Raqamli ta'lim texnologiyalari" kafedrası mudiri

Annotatsiya: Ushbu maqolada dasturlashda algoritmik yondashuvlarning turlari, ya'ni chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar nazariy jihatdan yoritilgan va ularning strukturaviy xususiyatlari tahlil qilingan. Har bir yondashuvning amaliy qo'llanilishi, dastur tuzishda tutgan o'rni, afzalliklari va qachon ishlatilishi kerakligi misollar bilan bayon etilgan.

Kalit so'zlar: Algoritm, chiziqli algoritm, tarmoqlanuvchi algoritm, takrorlanuvchi algoritm, tsikl, shart operatori, dasturlash, strukturaviy yondashuv, kompyuter fanlari.

Аннотация: В этой статье теоретически рассматриваются типы алгоритмических подходов к программированию, а именно линейные, разветвляющиеся и рекурсивные алгоритмы, и анализируются их структурные особенности. Практическое применение каждого подхода, его роль в программировании, его преимущества и когда его использовать, описываются на примерах.

Ключевые слова: Алгоритм, линейный алгоритм, разветвляющийся алгоритм, рекурсивный алгоритм, цикл, условный оператор, программирование, структурный подход, информатика.

Abstract: This article theoretically discusses the types of algorithmic approaches to programming, namely linear, branching, and recursive algorithms, and analyzes their structural features. The practical application of each approach,



its role in programming, its advantages, and when to use it are described with examples.

Keywords: Algorithm, linear algorithm, branching algorithm, recursive algorithm, cycle, conditional operator, programming, structural approach, computer science.

Zamonaviy kompyuter texnologiyalarining rivojlanishi bilan bir qatorda, dasturlash jarayonida algoritmlardan samarali foydalanish masalasi dolzarb ahamiyat kasb etmoqda. Algoritm — bu berilgan masalani hal qilish uchun ketma-ket bajariladigan aniq qadamlar to‘plamidir. Har qanday dastur asosida muayyan algoritmik struktura yotadi. Dasturchining vazifasi — masalaga mos keladigan algoritmik yondashuvni tanlab, uni samarali tarzda amalga oshirishdir.

Dasturlashda qo‘llaniladigan algoritmik yondashuvlar uch asosiy turga bo‘linadi: **chiziqli (ketma-ketlik)**, **tarmoqlanuvchi (shartli)** va **takrorlanuvchi (tsiklik)** tuzilmalar. Ushbu maqolada biz algoritmik yondashuvlarning nazariy asoslari, ularning strukturaviy farqlari, hamda real dasturlash muammolarida qo‘llanilishini analitik tarzda ko‘rib chiqamiz.

1. Chiziqli algoritmlar

Chiziqli algoritmlar eng sodda algoritmik tuzilma hisoblanadi. Bunday algoritmlarda barcha amallar ketma-ketlikda, shartsiz bajariladi. Har bir buyruq aniq tartibda bajarilib boradi va keyingi buyruq faqat oldingisi bajarilgach ishga tushadi. Chiziqli algoritmlar oddiy masalalar uchun juda samarali bo‘lib, ular faqat natijani hisoblash yoki ma’lumotni chiqarish kabi sodda amallarni bajaradi.

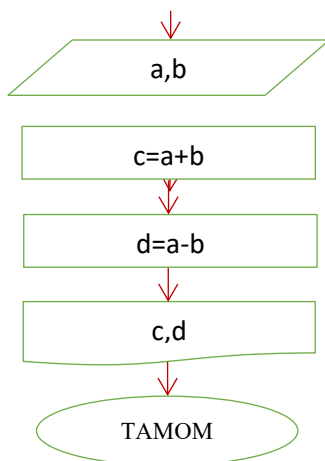
1-Misol Ikkita natural sonning yig‘indisi va ayirmasi hisoblovchi algoritm, blok – sxema tuzing va dasturini yozing.

Masalaning blok-sxemasi

BOSHLASH

Masalaning dasturiy yechimi

```
a = int(input("Birinchi natural sonni kiriting:"))
```



```
b = int(input("Ikkinchi natural sonni  
kiriting:"))
```

```
c = a + b
```

```
d = a - b
```

```
print("Yig'indisi:", c)
```

```
print("Ayirmasi:", d)
```

Natija: a=4, b=3 c=7, d=1

Masalaning algoritmi:

1. Boshlash;
2. b,c sonlarni qiymatini kiritish;
3. Yig'indini hisoblash: $c=a+b$;
4. Ayirmani hisoblash: $d=a-b$;
5. c va d natijani chiqarish;
6. Tugatish.

2. Tarmoqlanuvchi algoritmlar

Tarmoqlanuvchi algoritmlar shart asosida bir nechta yo'nalishlarda ishlash imkonini beradi. Bu turdagi algoritmlarda shart tekshiriladi va natijaga qarab tegishli yo'l tanlanadi. Asosan **if–else** strukturalari orqali ifodalanadi. Bu algoritmlar qaror qabul qilish jarayonlarida keng qo'llaniladi, ayniqsa foydalanuvchi kiritmalariga mos natija chiqarishda.

2-Misol. Birilgan x sonini musbat yoki manfiy son ekanligini aniqlovchi algoritmi, blok –sxema tuzing va dasturini yozing.

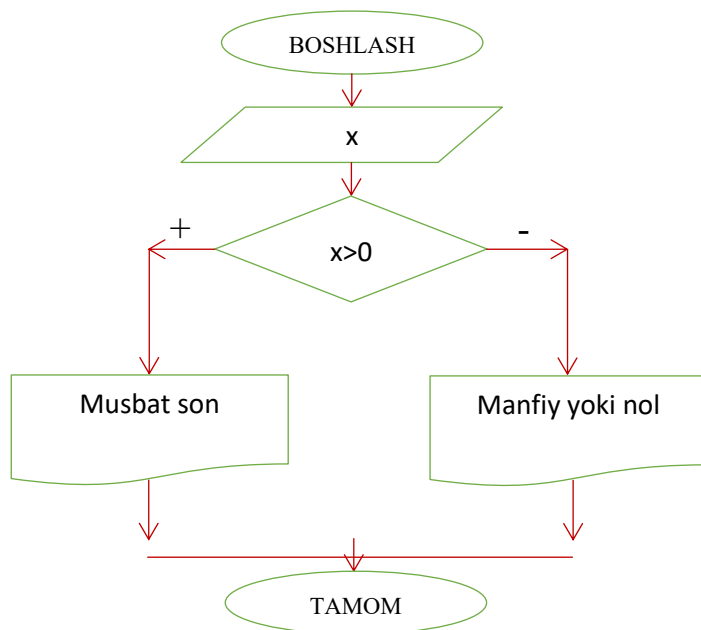
Masalaning algoritmi:

1. Boshlash;
2. x soni qiymatini kiritish;
3. Agar $x>0$ bo'lsa 4-bandga o'tsin, aks holda 5-bandga o'tsin;



4. Chop yetilsin “Musbat son” va 6-bandga o‘tsin;
5. Chop etilsin “Manfiy yoki nol” va 6-bandga o‘tsin;
6. Tugatish.

Masalaning blok-sxemasi



Masalaning dasturiy yechimi

```
x = int(input("Son kiriting: "))
```

```
if x > 0:
```

```
    print("Musbat son")
```

```
else:
```

```
    print("Manfiy yoki nol")
```

Natija: x=9 Musbat son

3. Takrorlanuvchi algoritmlar

Takrorlanuvchi algoritmlar muayyan amalni bir necha bor takrorlashni ta'minlaydi. Bunday algoritmlar **for**, **while** yoki **do-while** tsikllari orqali bajariladi. Tsikllar orqali massivlar, fayllar, ro'yxatlar ustida takroriy amallar bajariladi. Bu yondashuv katta hajmdagi ma'lumotlarni avtomatlashtirish uchun juda qulaydir.

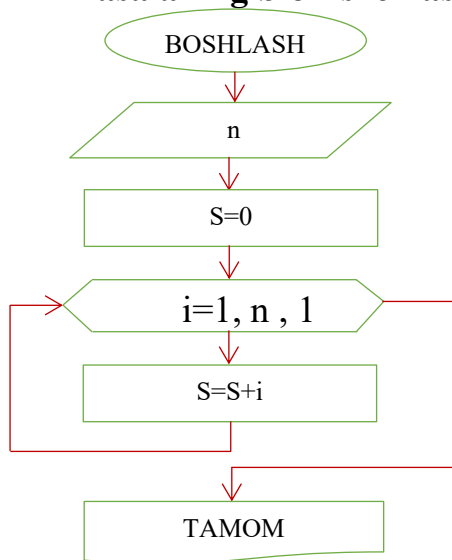
3-Misol. 1 dan n gacha bo'lgan natural sonlar yig'indisini hisoblovchi algoritmi, blok –sxema tuzing va dasturini yozing.

Masalaning algoritmi

1. Boshlash;
2. n soni qiymatini kiritish;
3. S=0 boshlang'ich qiymat berish;
4. i=1 dan n gacha takrorlash: S=S+i;
5. S yig'indi natijasini chiqarish;



6. Tugatish.

Masalaning blok-sxemasi**Xulosa**

Dasturlash jarayonida algoritmlarni to‘g‘ri tanlash va ulardan samarali foydalanish — muammoni hal etishning muhim qismidir. Chiziqli, tarmoqlanuvchi va takrorlanuvchi algoritmlar dastur asosidagi asosiy strukturalar bo‘lib, ular orqali har qanday murakkab masalani sodda va tushunarli qadamlar orqali yechish mumkin. Har bir algoritmi turi ma’lum bir masalaga mos tushadi. Masalan, oddiy hisoblashlar uchun chiziqli yondashuv yetarli bo‘lsa, murakkab qarorlar qabul qilish yoki katta hajmdagi takroriy amallarni bajarish uchun mos algoritmik struktura tanlash zarur bo‘ladi. Shu boisdan, dasturchilar uchun algoritmik yondashuvlarning nazariy va amaliy asoslarini chuqur o‘rganish zarurdir.

Masalaning dasturiy yechimi

```
n= int(input("n ni kiriting: "))
```

```
S = 0
```

```
for i in range(1, n + 1):
```

```
S += i
```

```
print("1 dan", n, "gacha sonlar  
yig'indisi:", S)
```

Natija: n=10 S=55

Foydalanilgan adabiyotlar:

1. Karimov A., Sodiqov S. *Algoritmlar va dasturlash asoslari*. – Toshkent: O‘zbekiston Milliy universiteti nashriyoti, 2021.
2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. *Introduction to Algorithms*. – MIT Press, 2009.
3. Paul Deitel, Harvey Deitel. *Python for Programmers*. – Pearson, 2020.



4. Abdulxayev A. *Dasturlash asoslari (Python misolida)*. – Toshkent: Innovatsiya nashriyoti, 2022.
5. Tanenbaum, A. S. *Structured Computer Organization*. – Prentice Hall, 2013.