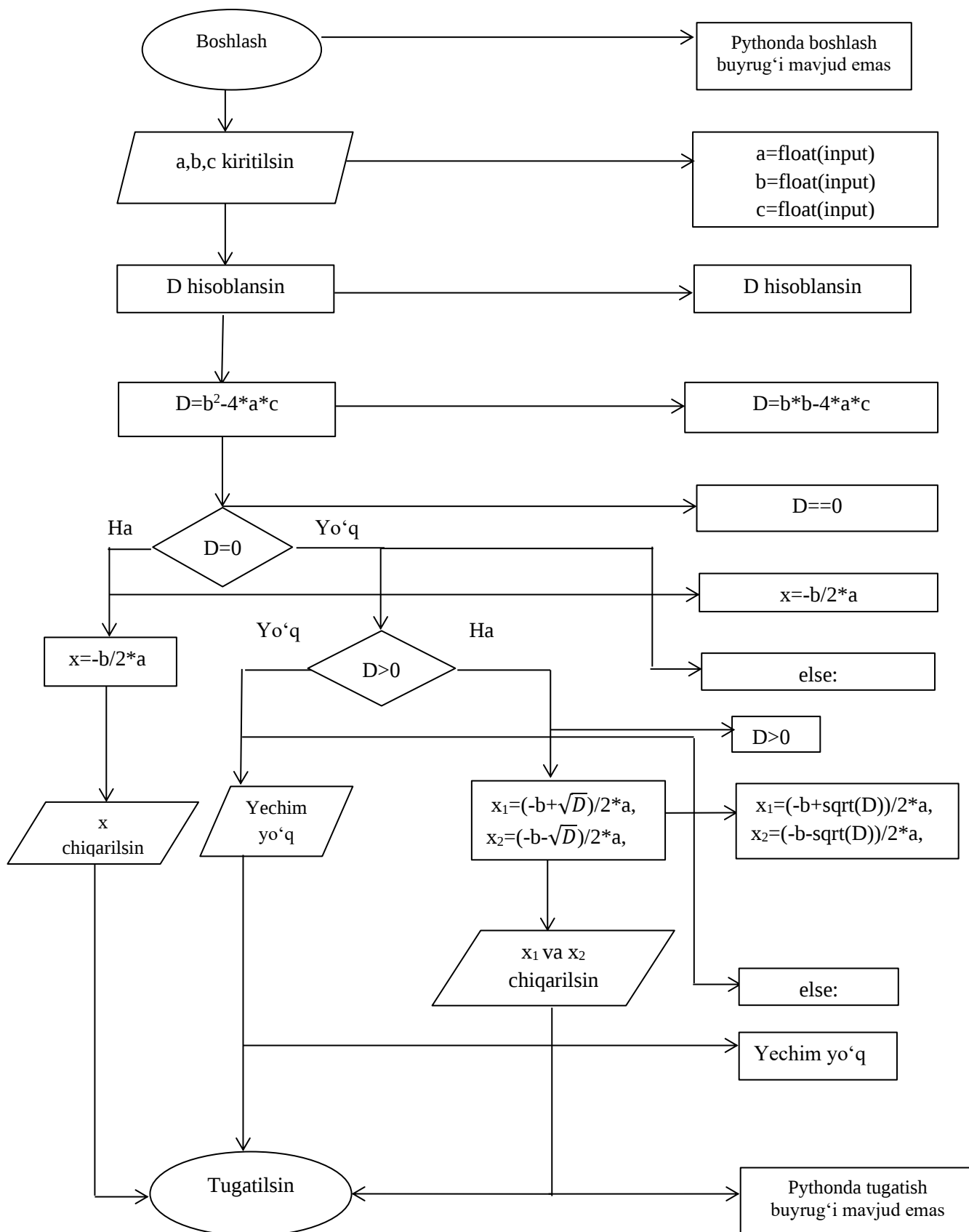


1-masala. Quyidagi masalani dasturlash:

$$ax^2 + bx + c = 0$$

Qaraliyotgan masala tarmoqlanuvchi algoritmi sirasiga kiradi. Ushbu algoritmi (masalani) sodda va tez tushunib dastur tiliga o'tkazishimiz uchun algoritmlashning blok-sxema turini tanlashni tavsiya etamiz. Berilgan masalani algoritmini blok-sxema ko'rinishda tasvirlagandan so'ng, uni Python dasturlash tilining dastur kodiga o'tkazish uchun mental sxemadan foydalaniladi (1-rasmga qarang).



1-rasm. Mental sxemadan foydalanish

Ushbu mental sxemadan foydalanish orqali o'quvchilar mustaqil ravishda quyidagi dastur kodini hosil qilishga erishiladi:

```

import cmath

def solve_quadratic(a, b, c):
    if a == 0:
        if b == 0:
            if c == 0:
                return ("Hammasi haqiqiy — tenglama identik.",)
            else:
                return ("Tenglama yechimsiz.",)
        return (-c / b,)
    D = b * b - 4 * a * c
    sqrtD = cmath.sqrt(D)
    x1 = (-b + sqrtD) / (2 * a)
    x2 = (-b - sqrtD) / (2 * a)
    return (x1, x2)

def main():
    print("Quadratic solver:  $ax^2 + bx + c = 0$ ")
    try:
        a = float(input("a ni kiriting: "))
        b = float(input("b ni kiriting: "))
        c = float(input("c ni kiriting: "))
    except ValueError:
        print("Iltimos, son kiriting.")
        return
    result = solve_quadratic(a, b, c)

    if isinstance(result, tuple) and len(result) == 1 and isinstance(result[0], str):
        print(result[0])
    return

```

```
if len(result) == 1:

    print(f"x = {result[0]}")

    return

x1, x2 = result

def fmt(x):

    if abs(getattr(x, "imag", 0)) < 1e-12:

        return f"{x.real:.12g}"

    return f"{x:.12g}"

print("x1 =", fmt(x1))

print("x2 =", fmt(x2))

if __name__ == "__main__":

    main()
```