```python
import os, sys
sys.path.append(os.path.join(os.path.dirname(__file__), '../ch02'))

from program2_1 import Dvector
from program8_1 import func
from program8_3 import rk4

def main():
    a, b, y0 = 0.0, 1.0, 1.0
    eps = 10.0 ** -8.0
    N = 10 # 最大反復回数N

    print("分割数を入力してください--->", end="")
    n = int(input())

    y = Dvector(0, n)

    # アダムス法
    y = adams( y0, y, a, b, n, N, eps, func )

    # 結果の表示
    h = (b - a) / n # 刻み幅
    for i in range(n+1):
        print("x={:.6f} \t y={:.6f} ".format(a+i*h, y[i]))


# アダムス法
def adams(y0: float, y: Dvector, a: float, b: float, n: int, N: int, eps: float, f) -> Dvector:
    y = Dvector(0, n) # y[0,1,...n] の確保
    F = Dvector(0, 4) # F[0,1,...4] の確保
    h = (b - a) / n    # 刻み幅の設定

    # スタータ
    y = rk4( y0, y, a, b, n, f )
    x = a
    for i in range(4):
        F[i] = f(x, y[i])
        x += h

    # 反復計算
    for i in range(3, n):
        # アダムス・バッシュフォース法
        F[3] = f(x-h, y[i])
        yp = y[i] + h*(55.0*F[3] - 59.0*F[2] + 37.0*F[1] - 9.0*F[0]) / 24.0
        for j in range(1, N+1):
            # アダムス・ムルトン法
            F[4] = f(x, yp)
            y[i+1] = y[i] + h*(9.0*F[4] + 19.0*F[3] - 5.0*F[2] + F[1]) / 24.0
            if abs(y[i+1] - yp) < eps:
                break
            yp = y[i+1]
        for j in range(1, 5):
            F[j-1] = F[j] # F[i] の更新
        x += h

    return y


if __name__ == "__main__":
    main()
```