#include <iostream>

#include <algorithm>

//FirstComeFirstServe:

float FCFS(int iAnzahlProzesse, int paiAnzahlProzesse[]) {

float iWartezeit = 0.f;

int iCounter = 0;

for (int i = 0; i < iAnzahlProzesse; i++)

{

iCounter += paiAnzahlProzesse[i];

iWartezeit += iCounter;

}

iWartezeit = (float) iWartezeit / iAnzahlProzesse;

return iWartezeit;

}

//ShortestJobFirst:

float SJF(int iAnzahlProzesse, int paiAnzahlProzesse[]) {

//array sortieren

std::sort(paiAnzahlProzesse, paiAnzahlProzesse+iAnzahlProzesse);

// Shortest Job First = First Come First Serve mit Sortierten werten

return FCFS(iAnzahlProzesse, paiAnzahlProzesse);

}

int main()

{

int iAnzahlProzesse = 0;

//Anzahl der Prozesse angeben

std::cout << "Eingabe der Anzahl der Prozesse: ";

std::cin >> iAnzahlProzesse;

//Dynamisches Array je nach Anzahl der Prozesse erstellen

int\* paiAnzahlProzesse = new int[iAnzahlProzesse];

//die Prozesslängen eingeben

for (int i = 0; i < iAnzahlProzesse; i++)

{

std::cout << "Eingabe der Prozesslaengen: "<< std::endl; //printf("Eingabe der Prozesslaengen: ");

std::cin >> paiAnzahlProzesse[i]; //scanf("%d ", &piAnzahlProzesse[i]); fflush(stdin);

}

//Ausgabe und Berechnung der Schedulingalgorithmen

// printf("die Durchschnittliche Wartezeit ist: %f" ,FCFS(iAnzahlProzesse, paiAnzahlProzesse));

printf("die Durchschnittliche Wartezeit ist: %f" , SJF(iAnzahlProzesse, paiAnzahlProzesse));

}