

a)

Darbą atliko pirmo kurso, pirmos grupės, antro pogrupio Programų Sistemų studentas
Ugnius Teišerskis.

Atlikta antrojo namų darbo ketvirtoji užduotis, kurios sąlyga yra tokia: „Įvesti sveiką skaičių N. Įvesti N sveikų skaičių. Tarp įvestų skaičių rasti ir išvesti du skaičius artimiausius įvestų skaičių vidurkiui: vieną mažesnę už vidurkį, o kitą - didesnę.“

Šios programos kūrimo metu skyriau prioritetą funkcijų rašymui. Mano intencija buvo parašyti tokias funkcijas, jog panorėjus, jas galima būtų panaudoti bet kurioje kitoje programoje. Šiam procesui ir buvo skirta daugiausiai laiko; nors programą, atliekančią sąlygoje nurodytą funkciją, parašiau per maždaug pusę valandos kelios dienos prieš pristatymo filmavimą, kiekvieną dieną vis įsijungdavau peržiūrėti kodą ir vis atrasdavau ką pakeisti: parašyti paprasčiau, trumpiau ar tiesiog lengviau suprantamai. Kaip pavyzdys: iki pristatymo dienos mano programoje buvo realizuotas BubbleSort algoritmas masyvo elementu surikiavimui, tačiau galiausiai nusprendžiau gyliau pasidomėti algoritmų sudėtingumais, vykdymu laikais ir priėjau prie išvados, jog quicksort algoritmas būtų labiau tinkamas ne tik šioje programoje, bet ir daugeliui ateities programų, todėl nusprendžiau realizuoti jos veikimą savo programoje. Su šiuo algoritmu jau buvo tekę dirbti per mokomąsias užduotis, rašyti jos versijas tiek su rekursija, tiek be jos, todėl rašymas neužtruko ilgai, tačiau vis vien reikėjo prisiminti jos veikimo principą. Taip pat buvo tokių atvejų, kai kitą dieną įsijungęs kodą, ne iškart suprasdavau, ką mano prieš tai parašytas kodas reiškia – todėl stengiausi perrašyti labiau suprantamai. Šiam įgūdžiui rašyti aiškų kodą nuo šiol skirsiu daugiau dėmesio. Taigi, šio namų darbo metu supratau, jog parašyti veikiantį kodą specifinei programai neužtenka – reikia taip pat užtikrinti jog kodas parašytas tvarkingai, aiškiai ir kad kodo fragmentai galėtų būti pritaikyti įvairiems scenarijams kitose programose.

b)

	PSP	A&D	C	T&R	S	Kita	Viso
Skirta minučių	30	45	95	25	30	0	225

c)

Data	Nuo	Iki	Trukdžiai	Laikas	Veikla	Komentarai	B.	Vnt.
2024/10/31	19:40	19:45	-	5	S	-	-	-
2024/10/31	19:45	20:00	-	15	A&D	Apgalvota, kaip turėtų atrodyti algoritmas, kokios gali kilti problemos.	-	-
2024/10/31	20:00	20:30	-	30	C	Parašytas kodas, atliekantis užduoties sąlygą.	Parašytos kodo eilutės	132
2024/10/31	20:30	20:40	-	10	T&R	Kartais grąžina nekorektiškus rezultatus; iškart identifiukuota loginė problema.	-	-
2024/10/31	20:40	20:45	-	5	C	Loginė problema išspręsta.	Parašytos (koreguotos) eilutės	10
2024/10/31	20:45	20:50	-	5	T&R	Visada gaunami korektiški rezultatai	-	-
2024/11/01	18:00	19:00	Kambariokas – 10 Vakarienė - 20	30	C	Sukurta #define kiekvienai žinutei, išvedamai į terminalą, pertvarkytas kodas, kad jis veiktų su defined žinutėmis. Kodo fragmentai iš int main perkelti į atskiras funkcijas, kai kurios funkcijos pertvarkytos, jog galėtų be problemų būti pritaikytos kitose programose ir t.t...	Parašytos (koreguotos) kodo eilutės	60
2024/11/01	19:00	19:05	-	5	T&R	Viskas ok.	-	-
2024/11/02	01:00	01:30	-	30	A&D	Quicksort yra greitesnis už BubbleSort ir būtų naudinga jį gerai įvaldyti dėl ateities.	-	-
2024/11/02	01:30	02:00	-	30	C	Pakeičiau bubblesort į quicksort, įsigilinau į abiejų algoritmų veikimo principus, jų minusus bei plusus,	Parašytos kodo eilutės	40
2024/11/02	02:00	02:05	-	5	T&R	Viskas ok.	-	-
2024/11/02	02:05	02:35	-	30	PSP	a ir b dalys	Parašyti žodžiai	300

d)

Radimo data	Defekto numeris	Defekto tipas	Padarytas fazėje	Pašalintas fazėje	Taisymo laikas	Taisant defektą	Detalus defekto aprašas
2024/10/31	1	Loginė klaida	Projektavimas	Kodavimas	5	5	Programa negaudavo korektiško rezultato tuo atveju, kai visi įvesti skaičiai masyve yra lygūs, pvz.: {1, 1, 1, 1}. Parašiau funkciją, kuri patikrina, ar visi masyvo elementai yra lygūs. Jei taip – į terminalą išvedama, jog įvestas skaičių masyvas neturi sprendinių, nes visi jo elementai yra lygūs, todėl nei vienas skaičius nėra nei mažesnis, nei didesnis už vidurkį.
2024/11/01	2	Efektyvumo klaida	Kodavimas	Kodavimas	30	30	Iki šiol buvo realizuotas BubbleSort rikiavimo algoritmas, kuris yra labai neefektyvus ir beveik nenaudojamas praktikoje. Pasidomėjęs gyčiau apie rikiavimo algoritmus, realizavau quicksort algoritmo veikimą savo programoje.