Indeks	Zad1	Zad2	Zad3	Ocena
209856	-	-	-	**(brak zadań)**
218129	-	-	-	**(brak zadań)**
218133	-	-	-	**(brak zadań)**
218135	1) Sortowanie- za dużo porównań oraz należy porównać całe nazwisko, a nie tylko pierwszą literę. for(int i=0;i<ile;i++){ for(int j=0;j<ile-1;j++) { if(dane[j].Nazwisko[0]>dane[j+1].Nazwisko[0]) { pom=dane[j]; dane[j]=dane[j+1]; dane[j+1]=pom; } } } Powinno być: for(int i=0;i<ile;i++) { for(int j=0;j<ile-i-1;j++){ if(strcmp(dane[j].Nazwisko,dane[j+1].Nazwisko)==1){ pom=dane[j]; dane[j]=dane[j+1]; dane[j+1]=pom;}}} Ocena 5=	1)Sortowanie – ta sama uwaga, co w zad.1 2)Usuwanie: for(int i=c;i<**baza.ile**;i++) { baza.Dane[i].Ocena=baza.Dane[i+1].Ocena; baza.Dane[i].Wiek=baza.Dane[i].Wiek; strcpy(baza.Dane[i].Nazwisko, baza.Dane[i+1].Nazwisko); } Powinno byc zmienione na: for(int i=c;i<**baza.ile-1**;i++) { baza.Dane[i]=baza.Dane[i+1]; } Ocena 5-	1)Sortowanie – uwaga z zad 1 i 2 2)Usuwanie – uwaga z zad2 Ocena 5	5-
218138	Ocena 5	Przy usuwaniu powinno byc for(int =ktory;i<**baz.ile-1**;i++) baz.Dane[i]=baz.Dane[i+1]; a jest: for (int i=ktory;i<baz.ile;i++) baz.Dane[i]=baz.Dane[i+1]; ocena 5	Ta sama uwaga dotycząca usuwania Ocena 5	5

218141	Sortowanie: należy przekazać indeksy skrajnych elementów, czyli ile-1, a nie ile. Powinno być: if (ile>1){ **Babelki(Dane, 0, ile-1);** --- Niepoprawna funkcja Zmien_ocene_osobie: dwukrotne przeszukiwanie tablicy oraz brak zwracania wskaźnika na znalezioną osobę: Jest: **Osoba* Zmien_ocene_osobie(float ocena, char * nazwisko, Osoba* Dane, int ile){** **if (Wyszukaj_wg_nazwiska(Dane, ile, nazwisko))** **{(*Wyszukaj_wg_nazwiska(Dane, ile, nazwisko)).Ocena = ocena;** **}** **else return NULL;** **}** Powinno być: Osoba* Zmien_ocene_osobie(float ocena, char * nazwisko, Osoba* Dane, int ile) { Osoba* pom; pom=Wyszukaj_wg_nazwiska(Dane, ile, nazwisko); if (pom!=NULL){ pom->Ocena = ocena; return pom; } else return NULL; } Ocena 4	Ta sama uwaga dotycząca sortowania Ocena 5	Ta sama uwaga dotycząca sortowania Należało przypisywać całe struktury, a nie ich składowe oraz ograniczyć liczbę alokacji: int Powieksz_tablice(Osoba*&Dane, int ile, int& rozmiar){ Osoba *local; rozmiar++; //local = new Osoba[ile]; **local=new Osoba[rozmiar];//dodano** for (int i = 0; i < ile; i++) { **local[i]=Dane[i];** } **delete[]Dane; //dodano** **Dane=local; //dodano** //Dane = new Osoba[rozmiar]; //for (int i = 0; i < ile; i++) //{ // **Dane[i]=local[i];** //} //delete []local; //local = 0; if (Dane)return 1; else return 0; } Ocena 5=	4.5
218160	-	-	-	**(brak zadań)**
218172	-	-	-	**(brak zadań)**
218182	-	-	-	**(brak zadań)**

218183	-	-	-	**(brak zadań)**
218194	-	-	-	**(brak zadań)**
	Ocena 4 (termin)	**Ocena 4 (termin)**	**1)Funkcja Powieksz_tablic jest nieprawidłowa:** **Powinno być:** -utworzenie nowej większej tablicy za pomocą new; jeśli uzyskano wartość NULL, ponieważ nie udało się utworzyc nowej tablicy, zwracana jest wartość 0 i kończy się funkcja --przeniesienie elementów ze starej tablicy do nowej tablicy, -usunięcie z pamięci starej tablicy za pomocą delete - przypisanie do wskaźnika starej tablicy wskaźnika nowej tablicy i zwrócenie wartości 1 przez funkcje i zakończenie funkcji 2) Podczas kopiowania zawartości tablic powinno być w funkcji Powieksz_tablice: **for(int i=0; i<ile; i++)** **new_tab[i]=Dane[i];** **Obecnie jest:** **for(int i=0; i<ile; i++)** **{** **new_tab[i].Wiek=Dane[i].Wiek;** **new_tab[i].Ocena=Dane[i].Ocena;** **strcpy(new_tab[i].Nazwisko,Dane[i].Nazwisko);** **}** 3)Obecnie bez żadnej kontroli pobiera się element typu Osoba o indeksie równym 0 w funkcji Powieksz_tablice Ocena 4	**4**
218250	-	-	-	**(brak**

				zadań)
218261	-	-	-	**(brak zadań)**
218283	Ocena 5	Usuwanie: jest for (; wsk<(baza.Dane + N); wsk++) *wsk = *(wsk + 1); Powinno być: for (; wsk<(baza.Dane +baza.ile-1); wsk++) *wsk = *(wsk + 1); Ocena 5	Ta sama uwaga dotycząca usuwania Ocena 5	5
218316	-	-	-	**(brak zadań)**
218319	1)W funkcji Dodaj_do_tablicy należało wykorzystac funkcję Wyszukaj_wg_nazwiska. Jest: **for (int i = 0; i<ile; i++) {** **if (!(strcmp(Nazwisko, Dane[i].Nazwisko))) {** **return false;** **}** **}** Dane[ile].Wiek = Wiek; Dane[ile].Ocena = Ocena; strcpy(Dane[ile].Nazwisko, Nazwisko); ile++; return true; Powinno być: **if(Wyszukaj_wg_nazwiska(Dane,ile,Nazwisko)!= NULL)** **return false;** Dane[ile].Wiek = Wiek;	1)Dodaj_do_tablicy – uwaga podobnie jak w zad1 2)Usuwanie: Jest for (int i = ktory; **i<baza.ile;** i++) baza.Dane[i]=baza.Dane[i+1]; Powinno być: for (int i = ktory; **i<baza.ile-1;** i++) baza.Dane[i]=baza.Dane[i + 1]; Ocena 5-	Uwagi takie jak w zad1 i 2 Ocena 5-	5-

	Dane[ile].Ocena = Ocena; strcpy(Dane[ile].Nazwisko, Nazwisko); ile++; return true; Ocena 5=			
218331	1)Przykład „brudnego kodu”, identycznego jak u studenta o numerze **218141**: Osoba* Zmien_ocene_osobie(float ocena, char * nazwisko, Osoba* Dane, int ile) { if(Wyszukaj_wg_nazwiska(Dane,ile,nazwisko)!=NULL) { Wyszukaj_wg_nazwiska(Dane,ile,nazwisko)->Ocena=ocena; } else return NULL; } Niedopuszczalne dwukrotne przeszukiwanie zbioru obiektów typu Osoba oraz brak instrukcji return. Dla 218141 podano sposób poprawy 2) Brak prototypów funkcji Ocena 4.5	1)Przy usuwaniu powinno byc for(int=ktory;i<**baza.ile-1**;i++) baza.Dane[i]=baza.Dane[i+1]; a jest: for(inti=ktory;i<baza.ile;i++) baza.Dane[i]=baza.Dane[i+1]; 2) Brak prototypów funkcji Ocena 5=	1)Brak prototypów funkcji 2) Usuwanie: jest for(int i=ktory;**i<ile**;i++) { Dane[i]=Dane[i+1]; Powinno być: for(int i=ktory;**i<ile-1**;i++) { Dane[i]=Dane[i+1]; Ocena 5=	5=
218351	-	-	-	**(brak zadań)**
218356	Nieprawidłowe sortowanie (algorytm wykonuje nadmiarowe porównania, można struktury przypisywać tzn można wykonać swap(dane[i-1],dane[i]);) Jest: for(int j=0; j<ile; j++) { **for(int i=1; i<ile; i++)** **{** **if((int)dane[i-1].nazwisko[0]>** **(int)dane[i].nazwisko[0])** **swap(dane[i-1].nazwisko,dane[i].nazwisko);** **}** **}**	Nieprawidłowe sortowanie (algorytm wykonuje nadmiarowe porównania, można struktury przypisywać tzn można wykonać: swap(baza.dane[i-1],baza.dane[i]);) Podczas wprowadzania nowej osoby nie sprawdza się, czy już taka osoba istnieje wg nazwiska – brakuje wywołania funkcji wyszukującej wg	Brak zadania W funkcjach: dodaj_osobe, zmien_oc należy zastosowac funkcję wyszukaj_nazw – dobry program charakteryzuje się wielokrotnym użyciem tego samej funkcji w tym samym kontekście. Ocena 4.5	4.5

	Powinno być: for(int j=0; j<ile-1; j++) { for(int i=0; **i<ile-j-1;** i++){ **if((strcmp(dane[i].nazwisko, dane[i+1].nazwisko)==1))** **swap(dane[i], dane[i+1]);** } } --- Podczas wprowadzania nowej osoby nie sprawdza się, czy już taka osoba istnieje wg nazwiska – brakuje funkcji wyszukującej wg nazwiska osobę dwa razy użytej: podczas wprowadzania danych i wyszukanie osoby do zmiany oceny. Brakuje prototypów funkcji Ocena 4	nazwiska osobę dwa razy użytej: podczas wprowadzania danych i wyszukanie osoby do zmiany oceny. Jest jedynie wywołana przy usuwaniu osoby Ocena 5=		
218361	1) Funkcja Podaj_tablice_znakow zawiera dwa razy użyty podobny kod do sprawdzenia zawartości łańcucha znaków. Należało wykonać jedną funkcję i dwa razy ją użyć. Jest to kod do usuwania spacji oraz jednakowych liter Ocena 5	1)Wstawianie łańcuchów- ta sama uwaga 2)Przy usuwaniu powinno być: for(int i=ktory;i<**baza.ile-1**;i++) { baza.Dane[i]=baza.Dane[i+1]; } A jest: for(int i=ktory;i<**baza.ile**;i++) { baza.Dane[i]=baza.Dane[i+1]; } Ocena 5	1)Wstawianie łańcuchów- ta sama uwaga 2)Usuwanie – uwaga taka sama jak w przypadku zad2 3)Podczas powiększania tablicy p**owinno być:** -utworzenie nowej większej tablicy za pomocą new, -przeniesienie elementów ze starej tablicy do nowej tablicy, -usunięcie z pamięci starej tablicy za pomocą delete - przypisanie do wskaźnika starej tablicy wskaźnika nowej tablicy 4)Przypisanie: ***&Dane=Danee;** **można zapisac:** **Dane=Danee** lub Osoba * Wykonaj_tablice(int& ile, int& rozmiar) {	5

			ile=0; rozmiar=1; Osoba *Dane = new Osoba[rozmiar]; **return Dane;** } Zamiast **return *&Dane;** Ocena 5=	
218368	-	-	-	**(brak zadań)**
218382	Ocena 5	1)Usuwanie: jest for (int i=ktory;**i<baza.ile**;i++) baza.Dane[i]=baza.Dane[i+1]; baza.ile--; powinno być: for (int i=ktory;**i<baza.ile-1**;i++) baza.Dane[i]=baza.Dane[i+1]; baza.ile--; ocena 5-	1)Usuwanie-podobna uwaga, jak w zad2 2)Powiększanie tablicy: **Powinno być:** -utworzenie nowej większej tablicy za pomocą new, -przeniesienie elementów ze starej tablicy do nowej tablicy, -usunięcie z pamięci starej tablicy za pomocą delete - przypisanie do wskaźnika starej tablicy wskaźnika nowej tablicy Ocena 4.5	5-
218387	-	-	-	**(brak zadań)**
218389	-	-	-	**(brak zadań)**
218390	-	-	-	**(brak zadań)**
	1) Niedopuszczalne trzy wyszukiwania w celu pokazania trzech atrybutów tej samej struktury case pokaz: { cout<<"Podaj szukane nazwisko: ";cin>>nazwisko;	**1) Ten program nie mógł poprawnie działac bez przypisanie na początku programu (podano na stronie 10 instrukcji do cw6):**	**1)Takie same uwagi jak w zad1. Konieczne do poprawy.** **Ocena 5**	**Ocena po poprawie/ 5**

	cout<<"Osoba szukana: "<< (*(**Wyszukaj_wg_nazwiska**(Dane, ile, nazwisko))).nazwisko; cout<<", wiek: "<<(*(**Wyszukaj_wg_nazwiska**(Dane, ile, nazwisko))).wiek; cout<<", ocena: "<< (*(**Wyszukaj_wg_nazwiska**(Dane, ile, nazwisko))).ocena<<endl; } Należy raz wyszukac obiekt i potem wyświetlić jego dane zgodnie z instrukcją w zad1 p.2 Np. cout<<"Podaj szukane nazwisko: ";cin>>nazwisko; Osoba* os= Wyszukaj_wg_nazwiska(Dane,ile,nazwisko); if(os!=NULL) { cout<<"Osoba szukana: "<<os->nazwisko; cout<<", wiek: "<<os->wiek; cout<<", ocena: "<<os->ocena; } } Należy ten kod przenieść do funkcji i wywołac ją w instrukcji switch. --- 2)Funkcja Zmien_ocene_osobie trzy razy wyszukuje obiekt typu Osoba, aby zmienic ocenę: Osoba * Zmien_ocene_osobie(float ocena, char nazwisko[], Osoba Dane[], int ile) { if(Wyszukaj_wg_nazwiska(Dane,ile,nazwisko)!=NULL) { (*(Wyszukaj_wg_nazwiska(Dane,ile,nazwisko))).ocena=ocena; return Wyszukaj_wg_nazwiska(Dane, ile,nazwisko); }; return NULL; } Ta definicja różni się od pomysłu kolegów 218331 i	**baza.ile=0;** --- **2)Takie same uwagi jak w zad1. Konieczne do poprawy.** --- 3)Usuwanie: jest for (int i=ktory;**i<baza.ile**;i++) baza.Dane[i]=baza.Dane[i+1]; baza.ile--; powinno być: for (int i=ktory;**i<baza.ile-1**;i++) baza.Dane[i]=baza.Dane[i+1]; baza.ile--; --- Ocena po poprawie Ocena 5		

	218141, że zostało dołożone trzecie wyszukiwanie: return Wyszukaj_wg_nazwiska(Dane, ile,nazwisko); Powinoo być: Osoba* Zmien_ocene_osobie(float ocena, char * nazwisko, Osoba* Dane, int ile) { Osoba* pom; pom=Wyszukaj_wg_nazwiska(Dane, ile, nazwisko); if (pom!=NULL){ pom->ocena = ocena; return pom; } else return NULL; } 2) Funkcja sort powinna miec jedynie postac: for(int l=0; l<ile-1; l++) for(int p=0; p<ile-1-l;p++) porownaj(t[p],t[p+1]); Obecnie złożoności obliczeniowa wzrosła z „ile do kwadratu" (typowa dla najwolniejszych algorytmów sortowania) do „ile do 4-j potęgi" (ile*ile*ile*ile), co jest niedopuszczalne **3)** Wyswielanie zawartości tablicy przeniesc do funkcji I wywoływac ją w case wypisz **Zadanie do poprawy!!!!** **Ocena 5**			
218394	-	-	-	**(brak zadań)**
218417	-	-	-	**(brak zadań)**
	4.5 (termin)	1) Przy usuwaniu brakuje zmniejszenia liczby elementów 2) Ocena 4(termin)	1)Przy powiększaniu tablicy należy usunąc niepotrzebną tablicę: for(int i = 0; i < ile; i++){ dane[i] = p[i]; }	**4**

			delete p;//dodano return true; ocena 4 (termin)	
218418	1) Funkcje: Dodaj_do_tablicy, Zmien_ocene_osobie, Wyswietl_osobe – złe warunki pętli: jest for(int a=0; **a<=ile**;a++) powinno byc: for(int a=0; **a<ile**;a++) --- 2) Funkcja Zmien_ocene ma dwa powtarzajace się wywolania funkcji Zmien_ocene_osobie oraz trzy przeszukania tablicy Dane– dwa w Zmien_ocene_osobie oraz trzecie w funkcji Wyswietl_osobe: Jest: **Zmien_ocene_osobie(Ocena, Nazwisko, Dane, ile);** **if (Zmien_ocene_osobie(Ocena, Nazwisko, Dane, ile)==0)** **Wyswietl_osobe (Dane, ile, Nazwisko);** W treści zadania podałam, że: **Osoba* osoba = Zmien_ocene_osobie(Ocena, Nazwisko, Dane, ile);** **if (osoba!=NULL)** **Wyswietl_osobe (*osoba);** Brakuje w zadaniu wieloużywalnej funkcji, która wyszukuje osobę i jest używana dwa razy: podczas dodawania nowej osoby oraz przy zmianie oceny --- 3)Sortowanie: wartosc p powinna być równa ile-1 Czyli prawidłowe wywołanie funkcji Babelki: **Babelki(Dane, ile-1);** A jest **Babelki(Dane, N);,** gdzie N jest rozmiarem tablicy, a powinno być	**Brak programu - niezbędny do zaliczenia całego ćwiczenia**	**Zadanie można potraktowac jako jedynie rozszerzenie zad1 – brakuje rozwiązania w postaci dynamicznej tablicy elementów typu Osoba** 1) Uwagi dotyczące sortowania z zad1, zmiany oceny oraz używania zmiennej ile –takie same dotyczą zadania 3 --- 2)Funkcja Usun_z_tablicy Należy przypisywac całe struktury: Jest for(;a<ile;a++) { **Dane[a].Wiek=Dane[a+1].Wiek;** **Dane[a].Ocena=Dane[a+1].Ocena;** **strcpy((Dane[a].Nazwisko), (Dane[a+1].Nazwisko));** } Powinno być: for(;a<ile;a++) **Dane[a]=Dane[a+1];** --- W funkcjach należy oddzielać przetwarzanie od prezentacji danych, czyli wstawiania danych z klawiatury I wyświetlania na ekran Np. int Usun_z_tablicy(Osoba Dane[], int & ile, char Nazwisko[]) { int a; for(a=0; a<=ile;a++) { if(strcmp(Dane[a].Nazwisko, Nazwisko)==0) { **/*cout<< "Nazwisko: "<<Dane[a].Nazwisko<<", Wiek: "<<Dane[a].Wiek<<",Ocena: <<Dane[a].Ocena<<endl;*/**	Brak zaliczenia

	indeksem ostatnio wstawionego elementu do tablicy, czyli ile-1 Wtedy należy poprawic: for(**int i = 1**; i<p; i++) for (int j = 0; j<p-i; j++) na for(**int i = 0**; i<p; i++) for (int j = 0; j<p-i; j++) Ocena 3.5		for(;a<ile;a++) { **/* Dane[a].Wiek=Dane[a+1].Wiek;** **Dane[a].Ocena=Dane[a+1].Ocena;** **strcpy((Dane[a].Nazwisko),** **(Dane[a+1].Nazwisko));*/** Dane[a]=Dane[a+1]; } ile--; break; } } Ocena 3.5 – jest to ocena za zad1	
218429	-	-	-	**(brak zadań)**
223556				5.5