

Indeks	Zad1	Zad2	Zad3	Ocena
209856	-	-	-	(brak zadań)
218129	-	-	-	(brak zadań)
218133	-	-	-	(brak zadań)
218135	<p>1) Sortowanie – zbyt wiele pętli</p> <pre>for(int i=0;i<ile;i++) {for(int j=0;j<ile-1;j++) { if(t[j][0]>t[j+1][0]) { strcpy(pom,t[j]); strcpy(t[j],t[j+1]); strcpy(t[j+1],pom);}}</pre> <p>Powinno być:</p> <pre>for(int i=0;i<ile-1;i++) {for(int j=0;j<ile-1-i;j++){ if(t[j][0]>t[j+1][0]) { strcpy(pom,t[j]); strcpy(t[j],t[j+1]); strcpy(t[j+1],pom);}}</pre> <hr/> <p>2)Usuwanie</p> <p>Jest:</p> <pre>for(int i=ktory;i<ile;i++)</pre> <p>Powinno być:</p> <pre>for(int i=ktory;i<ile-1;i++)</pre>	<p>1)Usuwanie – zalecana taka sama korekta ile na ile-1</p> <p>2) Sortowanie: źle dobrane indeksy w pętlach (jak w zad1)</p> <p>3)Należy przestawiac elementy tablicy, które są wskaźnikami na dynamiczne łańcuchy oraz porównac cały łańcuch znaków, a nie pierwszy znak łańcucha.</p> <p>Jest:</p> <pre>char pom[DL]; for(int i=0;i<ile;i++) { for(int j=0;j<ile-1;j++){ if(t[j][0]>t[j+1][0]) { strcpy(pom,t[j]); strcpy(t[j],t[j+1]); strcpy(t[j+1],pom); }}</pre> <p>Powinno być:</p> <pre>char* pom; for(int i=0;i<ile-1;i++) { for(int j=0;j<ile-1-i;j++) if(strcmp(t[j],t[j+1])!=1) { pom=t[j];</pre>	<p>1)Sortowanie jest niepoprawne-bąbelkowe sortowanie wymaga dwóch zagnieżdżonych pętli !!!</p> <p>2)Podczas sortowania należy przestawiac wskaźniki, a nie kopiowac składowe struktury. Powinno byc</p> <p>Dane[i]=Dane[i+1], gdzie elementy tablicy Dane są wskaźnikami!!!</p> <p>A porównuje się wybrane składowe struktury.</p> <p>2)Usuwanie – zalecana taka sama korekta ile na ile-1</p> <hr/> <p>Ocena 3.5</p>	4

	Ocena 4.5	<pre>t[j]=t[j+1]; t[j+1]=pom; }}}</pre> <hr/> Ocena 4.5		
218138	<p>1)Usuwanie for(int i=a;i<ile-1;i++) strncpy(*(Dane+i),*(Dane+i+1)); Jeśli ile jest równe N, oznacza to wyjście poza tablice dla indeksu równego i+1, czyli równe ile, czyli równe N Należy wstawić bezpośrednio na pozycję ile-1 pusty łańcuch.</p> <hr/> Ocena 5-	<p>1)Usuwanie-powinno być: for(int i=a;i<ile-1;i++) Dane[i]=Dane[i+1];</p> <hr/> <p>1)Sortowanie: powinno być inline void zamien(char* &a, char* &b) { char* pom; pom=a; a=b; b=pom; } Funkcja porownaj_zamien jest poprawna, ale należy zmienić nagłówek funkcji porownaj_zamien(char* &a, char* &b)</p> <hr/> Ocena 4.5	<p>1)Usuwanie –powinno być for(int i=a;i<ile-1;i++) Dane[i]=Dane[i+1]; ile--;</p> <hr/> <p>2)Sortowanie-należy przedstawiać wskaźniki na struktury: inline void zamien(Osoba* &a, Osoba* &b) { Osoba* pom = a; a = b; b = pom; } Nagłówek funkcji porównaj_zamien: inline void porównaj_zamien(Osoba* &a, Osoba* &b) { /*.....*/ }</p> <hr/> Ocena 4.5	4.5
218141	Ocena 5	<p>1)Sortowanie: powinno być inline void zamien(char* &a, char* &b) { char* pom;</p>	<p>1) Usuwanie z tablicy-brakuje usunięcia z pamięci usuwanej struktury: delete Dane[indeks];//dodano for (int i = indeks; i < ile - 1; i++)</p>	4.5

		<pre>pom=a; a=b; b=pom; }</pre> <p>Funkcja porownaj_zamien jest poprawna, ale należy zmienić nagłówek funkcji</p> <pre>porownaj_zamien(char* &a, char* &b)</pre> <hr/> <p>2)brakuje usunięcia z pamięci dynamicznego łańcucha podczas usuwania z tablicy:</p> <p>delete Dane[indeks];//dodano</p> <pre>for (int i = indeks; i < ile - 1; i++) {*(Dane + i)=*(Dane + i + 1);}</pre> <hr/> <p>Ocena 4.5</p>	<pre>{ *(Dane + i) = *(Dane + i + 1); }</pre> <hr/> <p>2) Sortowanie; Należy przestawić wskaźniki na struktury, a nie struktury:</p> <p>inline void zamien(Osoba* &a, Osoba* &b)</p> <pre>{ Osoba* pom = a; a = b; b = pom;}</pre> <p>Nagłówek funkcji porównaj_zamien:</p> <p>inline void porownaj_zamien(Osoba* &a, Osoba*&b)</p> <p>itd</p> <hr/> <p>Ocena 4.5</p>	
218160	Należy zmienić operacje na znakach na operacje na łańcuchach!!! W celu zaliczenia zadanie wg instrukcji do cw 7	brak	brak	Zaliczenie po uzupełnieniu zadań
218172	-	-	-	(brak zadań)
218182	-	-	-	(brak zadań)
218183	-	-	-	(brak

				zadań)
218194	-	-	-	(brak zadań)
	4.5 (termin)	<p>Program przetwarza tablicę wskaźników, dlatego należało posługiwać się wskaźnikami w operacjach na tablicy np.</p> <p>1)Usuwanie delete *(Dane+ktory); for(int l=ktory;l<ile-1;l++) { *(Dane+l)= *(Dane+(l+1)); } <hr/> 2)Sortowanie inline void zamien(char* &a, char* &b) { char*pom; pom=a; a=b; b=pom; } Funkcja porownaj_zamien jest poprawna, ale należy zmienić nagłówek funkcji porownaj_zamien(char* &a, char* &b) <hr/> Ocena 4</p>	<p>1)Sortowanie; Należy przestawiać wskaźniki na struktury, a nie struktury:</p> <p>inline void zamien(Osoba* &a, Osoba* &b){ Osoba* pom = a; a = b; b = pom;} Nagłówek funkcji porównaj_zamien: inline void porownaj_zamien(Osoba* &a, Osoba* &b) itd <hr/> Ocena 4</p>	4
218250	-	-	-	(brak zadań)
218261	-	-	-	(brak zadań)

218283	<p>1) return NULL nie powinno być podstawiane za int w funkcji Usun_z_tablicy</p> <p>2) swap powinna być używana dla struktur, a nie dla łańcuchów (w większości C++)</p> <hr/> <p>ocena 4</p>	<p>1) return NULL nie powinno być podstawiane za int w funkcji Usun_z_tablicy</p> <p>2) usuwanie powinno odbywać się za pomocą przestawiania referencji i należy zmienić warunek zakończenia pętli</p> <pre>delete *(Dane + ktory); for (; ktory < ile; ktory++) {memcpy(Dane + ktory, Dane + ktory + 1, DL); Czyli delete *(Dane + ktory); for (; ktory < ile-1; ktory++) {*(Dane + ktory) = *(Dane + ktory + 1);}</pre> <hr/> <p>3) wyznaczanie indeksów w pętlach sortujących powinny być zmienione na:</p> <pre>for (int i = 0; i < ile-1; i++) { for (int j = 0; j < ile - 1-i; j++)</pre> <hr/> <p>Ocena 4</p>	<p>1) należy zmienić warunek zakończenia pętli na:</p> <pre>for (; ktory < ile-1; ktory++)</pre> <p>2) wyznaczanie indeksów w pętlach sortujących powinny być zmienione na:</p> <pre>for (int i = 0; i < ile-1; i++) { for (int j = 0; j < ile - 1-i; j++)</pre> <hr/> <p>Ocena 5-</p>	4.5
218316	-	-	-	(brak zadań)
218319	-	-	-	(brak zadań)
	1) Podczas usuwania należy zmienić warunek zakończenia	1) Podczas sortowania należy przestawiać wskaźniki do	3) Podczas sortowania należy przestawiać wskaźniki do	4.5

	<p>pętli i<ile na i<ile-1 Ocena 4.5</p>	<p>dynamicznych łańcuchów, a nie kopiować łańcuchy!!!</p> <p>2) Podczas usuwania należało przestawić wskaźniki na łańcuchy oraz należy zmienić warunek zakończenia pętli i<ile na i<ile-1, czyli: for(int i = ktory; i<ile-1; i++){ *(Dane + i)=*(Dane + i + 1); } Ocena 4.5</p>	<p>struktur, a nie struktury!!! void zamien(Osoba* &a, Osoba* &b){ Osoba* c = a; a = b; b = c;} void porownaj_zamien(Osoba* &a, Osoba* &b){ if(strcmp(b->Nazwisko, a->Nazwisko) == -1) zamien(a, b); } 4) Podczas usuwania należy zmienić warunek zakończenia pętli i<ile na i<ile-1 Ocena 4.5 (termin)</p>	
218331	-	-	-	(brak zadań)
	<p>Brak prototypów funkcji 4 (termin)</p>	<p>Program przetwarza tablicę wskaźników, dlatego należało posługiwać się wskaźnikami w operacjach na tablicy np. 1)Brak prototypów funkcji. 2)Usuwanie-powinno być: delete *(Dane+ktory); for(int i=ktory;i<ile-1;i++) Dane[i]=Dane[i+1]; 3)Sortowanie inline void zamien(char* &a, char* &b) { char*pom;</p>	<p>1)Brak prototypów funkcji. 2)Usuwanie-powinno być: delete *(Dane+ktory); for(int i=ktory;i<ile-1;i++) Dane[i]=Dane[i+1];} 3)Sortowanie; Należy przestawiać wskaźniki na struktury, a nie struktury: inline void zamien(Osoba* &a, Osoba* &b){ Osoba* pom = a; a = b; b = pom;}</p>	4

		<pre>pom=a; a=b; b=pom; }</pre> <p>Funkcja porownaj_zamien jest poprawna, ale należy zmienić nagłówek funkcji</p> <pre>porownaj_zamien(char* &a, char* &b)</pre> <hr/> <p>Ocena 4-</p>	<p>Nagłówek funkcji porównaj_zamien:</p> <pre>inline void porownaj_zamien(Osoba* &a, Osoba*&b) itd</pre> <hr/> <p>Ocena 4</p>	
218351	-	-	-	(brak zadań)
218356	<p>1)Sortowanie-nieprawidłowo dobrane indeksy w funkcji sortuj. Powinno być:</p> <pre>void sortuj(char>(*Dane)[DL],int ile) { for(int i=0;i<ile-1;i++) {for (int j=0;j<ile-1-i;j++)</pre> <p>Należy porównać cały łańcuch za pomocą np. funkcji strcmp, natomiast w programie porównuje się pierwszy znak:</p> <pre>{if((*Dane+j)[0])>(int)*(Dane+j+1)[0]}</pre> <p>Funkcją swap można przestawiać struktury, natomiast nie tablice znaków:</p> <pre>swap(*(Dane+j),*(Dane+j+1));</pre>	<p>1)Sortowanie – należy zmienić indeksy w funkcji sortuj,</p> <hr/> <p>2) Funkcja usun : należy przestawiać wskaźniki!! I skorygować warunek wyjścia z pętli:</p> <pre>for (int i=indeks;i<ile-1;i++) *(Dane+i)=*(Dane+i+1);</pre> <hr/> <p>Nieprawidłowe działania na dynamicznych łańcuchach!!!</p> <p>Ocena 3</p>	<p>1) Źle dobrane indeksy w pętlach sortujących (poprawa jak w zad1)</p> <p>2) Indeksy przy usuwaniu powinny być zmienione</p> <pre>for (int i=indeks;i<ile;i++) { Dane[i]=Dane[i+1]; }</pre> <p>Powinno być:</p> <pre>for (int i=indeks;i<ile-1;i++) { Dane[i]=Dane[i+1]; }</pre> <hr/> <p>Ocena 4</p>	Ocena 3.5

	}} Ocena 3			
218361	Ocena 5	Ocena 5	1)Sortowanie-należy przedstawiać wskaźniki na struktury, a nie struktury inline void zamien(Osoba* &a, Osoba* &b){ Osoba* pom = a; a=b; b=pom;}	5
218368	-	-	-	(brak zadań)
218382	3) swap powinna być używana dla struktur, a nie dla łańcuchów (w większości C++) 2)Usuwanie powinno być: for (int i=ktory;i<ile-1;i++) strcpy(*(Dane+i),*(Dane+i+1)) ; Ocena 4	Zbyt wiele błędów	Zbyte wiele błędów	3-
218387	-	-	-	(brak zadań)
218389	-	-	-	(brak zadań)
218390	-	-	-	(brak zadań)
	1) Należy zmienić warunek pętli z:	brak	brak	zal

	<pre>for(int l=0; l<=ile-1; l++) for(int p=0; p<ile-l-1;p++) na: for(int l=0; l<ile-1; l++) for(int p=0; p<ile-l-1;p++) { Ocena 4 (termin)</pre>			
218394	-	-	-	(brak zadań)
218417	-	-	-	(brak zadań)
218418	-	-	-	(brak zadań)
218429	-	-	-	(brak zadań)
223556	5	5	5	5