

Sprawozdanie – Struktury, tablice statyczne i dynamiczne

Kacper Leporowski

W sprawozdaniu tym opisane zostaną zadania dotyczące wykorzystania statycznych i dynamicznych tablic, z wyjaśnieniami kodu i wynikiem dla przykładowych danych. Każde z zadań wykorzystuje strukturę o nazwie Student, w której przechowywane są dane o studencie.

```
struct Student
{
    char imie[15] = { '\0' };
    char nazwisko[30] = { '\0' };
    int indeks = NULL;
    char kierunek[30] = { '\0' };
    int rok = NULL;
    int grupa = NULL;
};
```

Zadanie 1

W pierwszym zadaniu wpierw deklarowana jest struktura typu Student, następnie za pomocą funkcji scanf do odpowiednich pól wczytywane są dane studenta – kolejno imię, nazwisko, indeks, kierunek, rok i grupa studenta. Dane te są potem wypisane w sposób sformatowany, przy użyciu funkcji printf.

```
struct Student Kacper;
scanf("%s %s %d %s %d %d", &Kacper.imie, &Kacper.nazwisko, &Kacper.indeks,
&Kacper.kierunek, &Kacper.rok, &Kacper.grupa);

printf("Imie: %s\nNazwisko: %s\nIndeks: %d\nKierunek: %s\nRok: %d\nGrupa: %d\n",
Kacper.imie, Kacper.nazwisko, Kacper.indeks, Kacper.kierunek, Kacper.rok, Kacper.grupa);
```

Wejście	Wyjście
Kacper Leporowski 151910 Infromatyka 1 8	Imie: Kacper Nazwisko: Leporowski Indeks: 151910 Kierunek: Infromatyka Rok: 1 Grupa: 8

Figura 1: Działanie programu 1

Zadanie 2

W tym zadaniu operacje wykonywane są nie na pojedynczej strukturze, a na statycznej tablicy struktur o rozmiarze 15. Wczytywanie i wypisywanie danych działa tak samo, jak w zadaniu pierwszym, lecz instrukcja umieszczona jest w pętli for, aby przejść przez wszystkie elementy tablicy.

W tym zadaniu dane studentów są dodatkowo sortowane – studenci uporządkowani są alfabetycznie według nazwiska, a w przypadku, gdy sprawdzana para studentów ma takie same nazwisko – sortowani są według imienia. Algorytm sortowania wykorzystany w programie to sortowanie bąbelkowe.

```
for (i = 0; i < 15; i++)
{
    for (j = i + 1; j < 15; j++)
    {
        struct Student temp;
        if (strcmp(ft_INF_8[i].nazwisko, ft_INF_8[j].nazwisko) > 0)
        {
            temp = ft_INF_8[i];
            ft_INF_8[i] = ft_INF_8[j];
            ft_INF_8[j] = temp;
        }
        else if (strcmp(ft_INF_8[i].nazwisko, ft_INF_8[j].nazwisko) == 0)
        {
            if (strcmp(ft_INF_8[i].imie, ft_INF_8[j].imie) > 0)
            {
                temp = ft_INF_8[i];
                ft_INF_8[i] = ft_INF_8[j];
                ft_INF_8[j] = temp;
            }
        }
        else
        {
            continue;
        }
    }
}
```

```
Jan Kowalski 12455 Informatyka 1 2
Michał Czerwinski 24356 Informatyka 2 4
Patryk Zajac 87533 Informatyka 1 5
Adrian Zajac 23457 Informatyka 1 3
Witold Adamczak 58025 Informatyka 1 8

-----
Imie: Witold
Nazwisko: Adamczak
Indeks: 58025
Kierunek: Informatyka
Rok: 1
Grupa: 8
-----
Imie: Michał
Nazwisko: Czerwinski
Indeks: 24356
Kierunek: Informatyka
Rok: 2
Grupa: 4
-----
Imie: Jan
Nazwisko: Kowalski
Indeks: 12455
Kierunek: Informatyka
Rok: 1
Grupa: 2
-----
Imie: Adrian
Nazwisko: Zajac
Indeks: 23457
Kierunek: Informatyka
Rok: 1
Grupa: 3
-----
Imie: Patryk
Nazwisko: Zajac
Indeks: 87533
Kierunek: Informatyka
Rok: 1
Grupa: 5
-----
```

Figura 2: Działanie programu dla przykładowych danych

Działanie programów z zadań 3 i 4 jest takie same, jak działanie programu z zadania 2, ale w nich wykorzystano tablice dynamiczne – w zadaniu 3 alokowane za pomocą funkcji charakterystycznych dla języka C, w zadaniu 4 za pomocą operacji dostępnych w języku C++.

Zadanie 3

Wpierw od użytkownika pobierana jest liczba całkowita, która ma być liczbą elementów w tablicy. Następnie za pomocą funkcji malloc alokowana jest pamięć dla tablicy dynamicznej – parametrem funkcji jest ilość bajtów zajmowana przez strukturę Student pomnożona przez liczbę wcześniej podaną przez użytkownika. Ze względu na to, że program został skompilowany za pomocą kompilatora języka C++, wynik funkcji jest dodatkowo rzutowany na typ Student – w przeciwnym wypadku kompilator uznawał liniijkę kodu za błędną.

```
struct Student* ft_INF_8 = (struct Student*)malloc(size * sizeof(*ft_INF_8));
```

Ponieważ zmienna o nazwie ft_INF_8 jest wskaźnikiem, sposób odwoływania się do wartości struktury różni się od sposobu wykorzystanego w zadaniu 2, co zostało pokazane na poniższym fragmencie kodu – zamiast odwoływać się do n-tego elementu tablicy, do zmiennej dodawana jest liczba n – w praktyce oznacza to dodanie liczby n do adresu, który zmienna zajmuje w pamięci, czyli tak naprawdę adresu pierwszego elementu tablicy. Do wydobywania konkretnej wartości ze struktury wykorzystuje się również strzałkę zamiast kropki.

```
//Odwołanie się do indeksu studenta - ujęcie statyczne
```

```
ft_INF_8[n].indeks = 0;
```

```
//Odwołanie się do indeksu studenta - ujęcie dynamiczne
```

```
(ft_INF_8+n)->indeks = 0;
```

Mając tą różnicę na uwadze, reszta operacji przebiega identycznie jak w zadaniu 2. Na końcu programu, w celu zwolnienia pamięci zajmowanej przez tablicę, wykorzystywana jest funkcja free.

```
free(ft_INF_8);
```

Wejście	Wyjście
Rozmiar tablicy: 5 Jan Nowak 12426 Informatyka 1 8 Miroslaw Brzezinski 87457 Informatyka 1 8 Marek Albecki 97456 Informatyka 1 8 Agnieszka Albecka 893746 Informatyka 1 8 Stefan Czyzyk 2345 Informatyka 1 8	----- Imie: Agnieszka Nazwisko: Albecka Indeks: 893746 Kierunek: Informatyka Rok: 1 Grupa: 8 ----- Imie: Marek Nazwisko: Albecki Indeks: 97456 Kierunek: Informatyka Rok: 1 Grupa: 8 ----- Imie: Miroslaw Nazwisko: Brzezinski Indeks: 87457 Kierunek: Informatyka Rok: 1 Grupa: 8 ----- Imie: Stefan Nazwisko: Czyzyk Indeks: 2345 Kierunek: Informatyka Rok: 1 Grupa: 8 ----- Imie: Jan Nazwisko: Nowak Indeks: 12426 Kierunek: Informatyka Rok: 1 Grupa: 8 -----

Figura 3: Działanie programu 3 dla przykładowych danych

Zadanie 4

Jedyna istotna różnica między zadaniem 3 a zadaniem 4 to sposób alokacji pamięci dla zmiennej – do alokacji wykorzystuje się słowo kluczowe `new`, a do usunięcia zmiennej słowo kluczowe `delete` (w przypadku tablicy należy pamiętać o dodaniu pary nawiasów kwadratowych).

```
struct Student* ft_INF_8 = new struct Student[size];  
//...  
delete [] ft_INF_8;
```

Wejście	Wyjście
Rozmiar tablicy: 5 Olgierd Zamachowski 98576 Informatyka 1 5 Maria Gorecka 12356 Informatyka 1 6 Dawid Biniarz 92861 Informatyka 1 7 Bartosz Kaczmarek 132464 Informatyka 1 8 Aleksandra Kaczmarek 234566 Informatyka 1 9	----- Imie: Dawid Nazwisko: Biniarz Indeks: 92861 Kierunek: Informatyka Rok: 1 Grupa: 7 ----- Imie: Maria Nazwisko: Gorecka Indeks: 12356 Kierunek: Informatyka Rok: 1 Grupa: 6 ----- Imie: Aleksandra Nazwisko: Kaczmarek Indeks: 234566 Kierunek: Informatyka Rok: 1 Grupa: 9 ----- Imie: Bartosz Nazwisko: Kaczmarek Indeks: 132464 Kierunek: Informatyka Rok: 1 Grupa: 8 ----- Imie: Olgierd Nazwisko: Zamachowski Indeks: 98576 Kierunek: Informatyka Rok: 1 Grupa: 5 -----

Figura 4: Działanie programu 4 dla przykładowych danych