

Sprawozdanie – programowanie niskopoziomowe (lab3)

Adam Mikołajczak 151247

1. Zaimplementowanie struktury student, następnie utworzenie struktury „Ja” – która ma reprezentować moje dane.

```
struct Student {  
    char imie[20];  
    char nazwisko[30];  
    int nr_indeksu;  
    char kierunek[30];  
    int rok_studiow;  
    int nr_gr_dziekanskiej;  
}Ja;
```

Wczytanie danych od użytkownika

```
printf("Podaj imie\n");  
scanf(" %s", Ja.imie);  
  
printf("Podaj nazwisko\n");  
scanf(" %s", Ja.nazwisko);  
  
printf("Podaj nr indeksu\n");  
scanf(" %d", &Ja.nr_indeksu);  
  
printf("Podaj kierunek\n");  
scanf(" %s", Ja.kierunek);  
  
printf("Podaj rok_studiow\n");  
scanf(" %d", &Ja.rok_studiow);  
  
printf("Podaj nr_gr_dziekanskiej\n");  
scanf(" %d", &Ja.nr_gr_dziekanskiej);
```

Wypisanie danych w czytelny sposób:

```
printf("\nImie: %s\n", Ja.imie);  
printf("Nazwisko: %s\n", Ja.nazwisko);  
printf("nr_indeksu: %d\n", Ja.nr_indeksu);  
printf("kierunek: %s\n", Ja.kierunek);  
printf("rok_studiow: %d\n", Ja.rok_studiow);  
printf("nr_gr_dziekanskiej: %d\n", Ja.nr_gr_dziekanskiej);
```

Input oraz output:

```
Podaj numer zadania  
1  
Podaj imie  
Adam  
Podaj nazwisko  
Mikołajczak  
Podaj nr indeksu  
151247  
Podaj kierunek  
informatyka  
Podaj rok_studiow  
1  
Podaj nr_gr_dziekanskiej  
8  
  
Imie: Adam  
Nazwisko: Mikołajczak  
nr_indeksu: 151247  
kierunek: informatyka  
rok_studiow: 1  
nr_gr_dziekanskiej: 8
```

2. Zdefiniowanie tablicy statycznej o rozmiarze 15

```
Student ft_INF_8[15]{};
```

Wczytanie 4 studentów do tablicy

```
for (int i = 0; i < 4; i++) {
    printf("Podaj imie\n");
    scanf(" %s", ft_INF_8[i].imie);

    printf("Podaj nazwisko\n");
    scanf(" %s", ft_INF_8[i].nazwisko);

    printf("Podaj nr indeksu\n");
    scanf(" %d", &ft_INF_8[i].nr_indeksu);

    printf("Podaj kierunek\n");
    scanf(" %s", ft_INF_8[i].kierunek);

    printf("Podaj rok_studiow\n");
    scanf(" %d", &ft_INF_8[i].rok_studiow);

    printf("Podaj nr_gr_dziekanskiej\n");
    scanf(" %d", &ft_INF_8[i].nr_gr_dziekanskiej);
}
```

Sortowanie bąbelkowe alfabetycznie ze względu na nazwisko oraz imię

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

Wypisanie danych

```
for (int i = 0; i < 15; i++) {
    if (!(strlen(ft_INF_8[i].imie) == 0)) {
        printf("Imie: %s\n", ft_INF_8[i].imie);
        printf("Nazwisko: %s\n", ft_INF_8[i].nazwisko);
        printf("nr_indeksu: %d\n", ft_INF_8[i].nr_indeksu);
        printf("kierunek: %s\n", ft_INF_8[i].kierunek);
        printf("rok_studiow: %d\n", ft_INF_8[i].rok_studiow);
        printf("nr_gr_dziekanskiej: %d\n", ft_INF_8[i].nr_gr_dziekanskiej);
    }
}
```

Dane wejściowe:

```
Podaj numer zadania
2
Podaj imie
Adam
Podaj nazwisko
Wawrzyniak
Podaj nr indeksu
151000
Podaj kierunek
informatyka
Podaj rok_studiow
1
Podaj nr_gr_dziekanskiej
8
Podaj imie
Adam
Podaj nazwisko
Bartkowiak
Podaj nr indeksu
151001
Podaj kierunek
informatyka
Podaj rok_studiow
1
Podaj nr_gr_dziekanskiej
1
Podaj imie
Bartek
Podaj nazwisko
Adamczak
Podaj nr indeksu
151002
Podaj kierunek
informatyka
Podaj rok_studiow
1
Podaj nr_gr_dziekanskiej
8
Podaj imie
Bartek
Podaj nazwisko
Zebrzynski
Podaj nr indeksu
151003
Podaj kierunek
informatyka
Podaj rok_studiow
1
Podaj nr_gr_dziekanskiej
8
```

Dane wyjściowe

```
Imie: Bartek
Nazwisko: Adamczak
nr_indeksu: 151002
kierunek: informatyka
rok_studiow: 1
nr_gr_dziekanskiej: 8
Imie: Adam
Nazwisko: Bartkowiak
nr_indeksu: 151001
kierunek: informatyka
rok_studiow: 1
nr_gr_dziekanskiej: 1
Imie: Adam
Nazwisko: Wawrzyniak
nr_indeksu: 151000
kierunek: informatyka
rok_studiow: 1
nr_gr_dziekanskiej: 8
Imie: Bartek
Nazwisko: Zebrzynski
nr_indeksu: 151003
kierunek: informatyka
rok_studiow: 1
nr_gr_dziekanskiej: 8
```

3. Tworzenie dynamicznej tablicy studentów

```
int wielkosc;
printf("Podaj wielkosc tablicy\n");
scanf("%d", &wielkosc);
Student *ft_INF_8;
ft_INF_8 = (Student*) malloc(sizeof(Student) * wielkosc);
```

Wczytanie 33% tablicy

```
for (int i = 0; i < (wielkosc / 3); i++) {
    printf("Podaj imie\n");
    scanf(" %s", ft_INF_8[i].imie);

    printf("Podaj nazwisko\n");
    scanf(" %s", ft_INF_8[i].nazwisko);

    printf("Podaj nr indeksu\n");
    scanf(" %d", &ft_INF_8[i].nr_indeksu);

    printf("Podaj kierunek\n");
    scanf(" %s", ft_INF_8[i].kierunek);

    printf("Podaj rok studiow\n");
    scanf(" %d", &ft_INF_8[i].rok_studiow);

    printf("Podaj nr_gr_dziekanskiej\n");
    scanf(" %d", &ft_INF_8[i].nr_gr_dziekanskiej);
}
```

Algorytm sortowania identyczny jak wyżej:

```
Student temp;
for (int i = 0; i < wielkosc; i++) {
    for (int j = 0; j < wielkosc - i - 1; j++) {

        if (strcmp(ft_INF_8[j].nazwisko, ft_INF_8[j + 1].nazwisko) == 0) {
            if (strcmp(ft_INF_8[j].imie, ft_INF_8[j + 1].imie) > 0) {
                memcpy(&temp, ft_INF_8 + j, sizeof(struct Student));
                memcpy(ft_INF_8 + j, ft_INF_8 + j + 1, sizeof(struct Student));
                memcpy(ft_INF_8 + j + 1, &temp, sizeof(struct Student));
            }
        }

        else if (strcmp(ft_INF_8[j].nazwisko, ft_INF_8[j + 1].nazwisko) > 0) {
            memcpy(&temp, ft_INF_8 + j, sizeof(struct Student));
            memcpy(ft_INF_8 + j, ft_INF_8 + j + 1, sizeof(struct Student));
            memcpy(ft_INF_8 + j + 1, &temp, sizeof(struct Student));
        }
    }
}
```

Wypisywanie wymagało innego warunku, które sprawdza, czy pierwszy znak imienia znajduje się w literach z tablicy ASCII

```
for (int i = 0; i < wielkosc; i++) {
    if ((ft_INF_8[i].imie[0] > 64 && ft_INF_8[i].imie[0] < 91) || (ft_INF_8[i].imie[0] > 96 && ft_INF_8[i].imie[0] < 123)) {
        //printf("%d", strlen(ft_INF_8[i].imie));

        printf("\nImie: %s\n", ft_INF_8[i].imie);
        printf("Nazwisko: %s\n", ft_INF_8[i].nazwisko);
        printf("nr_indeksu: %d\n", ft_INF_8[i].nr_indeksu);
        printf("kierunek: %s\n", ft_INF_8[i].kierunek);
        printf("rok_studiow: %d\n", ft_INF_8[i].rok_studiow);
        printf("nr_gr_dziekanskiej: %d\n", ft_INF_8[i].nr_gr_dziekanskiej);
    }
}
```

Wynik wypisywania identyczny jak wyżej

Zwolnienie pamięci za pomocą funkcji free

```
free(ft_INF_8);
```

4. Utworzenie tablicy metodą z C++

```
int wielkosc;  
printf("Podaj wielkosc tablicy\n");  
scanf("%d", &wielkosc);  
Student* ft_INF_8;  
ft_INF_8 = new Student[wielkosc];
```

Wczytanie 33% studentów

```
for (int i = 0; i < (wielkosc / 3); i++) {  
    printf("Podaj imie\n");  
    scanf(" %s", ft_INF_8[i].imie);  
  
    printf("Podaj nazwisko\n");  
    scanf(" %s", ft_INF_8[i].nazwisko);  
  
    printf("Podaj nr indeksu\n");  
    scanf(" %d", &ft_INF_8[i].nr_indeksu);  
  
    printf("Podaj kierunek\n");  
    scanf(" %s", ft_INF_8[i].kierunek);  
  
    printf("Podaj rok_studiow\n");  
    scanf(" %d", &ft_INF_8[i].rok_studiow);  
  
    printf("Podaj nr_gr_dziekanskiej\n");  
    scanf(" %d", &ft_INF_8[i].nr_gr_dziekanskiej);  
}
```

Algorytm sortowania, wypisywanie elementów, oraz dane wejściowe i wyjściowe takie jak wyżej

```
Student temp;  
for (int i = 0; i < wielkosc; i++) {  
    for (int j = 0; j < wielkosc - i - 1; j++) {  
  
        if (strcmp(ft_INF_8[j].nazwisko, ft_INF_8[j + 1].nazwisko) == 0) {  
            if (strcmp(ft_INF_8[j].imie, ft_INF_8[j + 1].imie) > 0) {  
                memcpy(&temp, ft_INF_8 + j, sizeof(struct Student));  
                memcpy(ft_INF_8 + j, ft_INF_8 + j + 1, sizeof(struct Student));  
                memcpy(ft_INF_8 + j + 1, &temp, sizeof(struct Student));  
            }  
        }  
  
        else if (strcmp(ft_INF_8[j].nazwisko, ft_INF_8[j + 1].nazwisko) > 0) {  
            memcpy(&temp, ft_INF_8 + j, sizeof(struct Student));  
            memcpy(ft_INF_8 + j, ft_INF_8 + j + 1, sizeof(struct Student));  
            memcpy(ft_INF_8 + j + 1, &temp, sizeof(struct Student));  
        }  
    }  
}  
  
for (int i = 0; i < wielkosc; i++) {  
    if ((ft_INF_8[i].imie[0] > 64 && ft_INF_8[i].imie[0] < 91) || (ft_INF_8[i].imie[0] > 96 && ft_INF_8[i].imie[0] < 123)) {  
        //printf("%d", strlen(ft_INF_8[i].imie));  
  
        printf("\nImie: %s\n", ft_INF_8[i].imie);  
        printf("Nazwisko: %s\n", ft_INF_8[i].nazwisko);  
        printf("nr_indeksu: %d\n", ft_INF_8[i].nr_indeksu);  
        printf("kierunek: %s\n", ft_INF_8[i].kierunek);  
        printf("rok_studiow: %d\n", ft_INF_8[i].rok_studiow);  
        printf("nr_gr_dziekanskiej: %d\n", ft_INF_8[i].nr_gr_dziekanskiej);  
    }  
}
```

Zwolnienie pamięci używając operacji delete

```
delete[] ft_INF_8;
```