Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 2

по дисциплине ОАиП

Вариант 3

Выполнил студент гр. 150502: Альхимович Н.Г.

Проверил:         Дулько П.А.

Минск 2022

Задание:

В командной строке задаются два признака. Первый признак определяет тип вводимой информации: просто структура или структура, включающая объединение и имена функций, которые надо выполнить. Функции выводят информацию (из просто структуры или из структуры, включающей объединение) в зависимости от второго признака в командной строке.

В структурах хранится информация о студентах: фамилия, имя, отчество. В объединении структура хранит один из типов информации:

а) отец, мать, брат;

б) отец, мать, брат, сестра;

в) мать, брат, сестра.

Листинг кода:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct student //объявление структуры

{

char \*surname;

char \*name;

char \*patronymic;

};

struct student\_add //объявление структуры, включающей объединение

{

char \*q1;

char \*q2;

union relatives

{

struct type\_a

{

char \*father;

char \*mother;

char \*brother;

} a;

struct type\_b

{

char \*father;

char \*mother;

char \*brother;

char \*sister;

} b;

struct type\_c

{

char \*mother;

char \*brother;

char \*sister;

} c;

} rel;

char \*type; //переменная, указывающая на тип хранимой в данный момент структуры

};

int main(int argc, char \*\*argv)

{

int n;

struct student \*ptr1 = NULL; //указатель на структуру

struct student\_add \*ptr2 = NULL; //указатель на структуру с объединением

void input1(int n, struct student \*ptr1); //прототип функции ввода структуры

void output1(int n, struct student \*ptr1); //прототип функции вывода информации из структуры

void input2(int n, struct student\_add \*ptr2); //прототип функции ввода структуры с объединением

void output2(int n, struct student\_add \*ptr2); //прототип функции выводы информации структуры с объединением

printf("Введите количество студентов: ");

while((scanf("%d", &n)) != 1)

{

printf("\nНекорректный ввод, повторите попытку: ");

scanf("%d", &n);

fflush(stdin);

}

// printf("\nЕсли необходим список всех студентах, введите 1; если нужна дополнительная информация о родственниках, введите 2");

//

// printf("\nЕсли Вы ввели 1, то теперь введите Output1, в противном случае - Output2");

if(\*argv[1] == '1') //если необходимо работать с обычной структурой

{

if((ptr1 = (struct student \*) calloc(n, sizeof(struct student))) == NULL)

{

printf("\nПамять не выделена");

return 0;

}

input1(n, ptr1);

}

else //если необходимо работать со структурой с объединением

{

if((ptr2 = (struct student\_add \*) calloc(n, sizeof(struct student\_add))) == NULL)

{

printf("\nПамять не выделена");

return 0;

}

input2(n, ptr2);

}

if(\*argv[2] == '1')

output1(n, ptr1);

else

output2(n, ptr2);

printf("\n");

return 0;

}

void input1(int n, struct student \*ptr1)

{

int i;

for(i=0; i<n; i++) //ввод ФИО каждого студента

{

printf("\nВведите данные о %d-ом студенте: ", i+1);

printf("\nВведите фамилию: ");

if((ptr1[i].surname = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr1[i].surname, 21, stdin);

printf("Введите имя: ");

if((ptr1[i].name = (char \*) calloc(10, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr1[i].name, 11, stdin);

printf("Введите отчество: ");

if((ptr1[i].patronymic = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr1[i].patronymic, 21, stdin);

}

}

void output1(int n, struct student \*ptr1)

{

int i;

printf("\nИнформация о студентах: ");

printf("\nНомер ФИО");

for(i=0; i<n; i++) //вывод ФИО студентов

{

printf("\n%-3d%s%s%s", i+1, ptr1[i].surname, ptr1[i].name, ptr1[i].patronymic);

}

for(i=0; i<n; i++) //освобождение памяти, выделенной под указатели в структуре

{

free(ptr1[i].name);

free(ptr1[i].surname);

free(ptr1[i].patronymic);

}

free(ptr1); //освобождение памяти, выделенной под указатель на структуру

}

void input2(int n, struct student\_add \*ptr2)

{

int i;

for(i=0; i<n; i++) //ввод информации о родственниках студентов

{

printf("\nВведите данные о %d-ом студенте: ", i+1);

printf("\nЕсли у студента полная семья, введите yes; иначе - no: "); //параметры, определяющие тип структуры из объединения

if((ptr2[i].q1 = (char \*) calloc(3, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].q1, 4, stdin);

printf("Если у студента есть и брат, и сестра, введите yes; иначе - no: ");

if((ptr2[i].q2 = (char \*) calloc(3, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].q2, 4, stdin);

if(memcmp(ptr2[i].q1, "yes", 2) == 0 && memcmp(ptr2[i].q2, "no", 2) == 0) //если семья полная

{

printf("\nВведите фамилию и инициалы отца: ");

if((ptr2[i].rel.a.father = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.a.father, 21, stdin);

printf("Введите фамилию и инициалы матери: ");

if((ptr2[i].rel.a.mother = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.a.mother, 21, stdin);

printf("Введите фамилию и инициалы брата: ");

if((ptr2[i].rel.a.brother = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.a.brother, 21, stdin);

if((ptr2[i].type = (char \*) calloc(1, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

ptr2[i].type = "a";

}

else if(memcmp(ptr2[i].q1, "yes", 2) == 0 && memcmp(ptr2[i].q2, "yes", 2) == 0) //если семья многодетная

{

printf("\nВведите фамилию и инициалы отца: ");

if((ptr2[i].rel.b.father = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.b.father, 21, stdin);

printf("Введите фамилию и инициалы матери: ");

if((ptr2[i].rel.b.mother = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.b.mother, 21, stdin);

printf("Введите фамилию и инициалы брата: ");

if((ptr2[i].rel.b.brother = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.b.brother, 21, stdin);

printf("Введите фамилию и инициалы сестры: ");

if((ptr2[i].rel.b.sister = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.b.sister, 21, stdin);

if((ptr2[i].type = (char \*) calloc(1, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

ptr2[i].type = "b";

}

else if(memcmp(ptr2[i].q1, "no", 2) == 0 && memcmp(ptr2[i].q2, "yes", 2) == 0) //если семья неполная

{

printf("\nВведите фамилию и инициалы матери: ");

if((ptr2[i].rel.c.mother = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.c.mother, 21, stdin);

printf("Введите фамилию и инициалы брата: ");

if((ptr2[i].rel.c.brother = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.c.brother, 21, stdin);

printf("Введите фамилию и инициалы сестры: ");

if((ptr2[i].rel.c.sister = (char \*) calloc(20, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(ptr2[i].rel.c.sister, 21, stdin);

if((ptr2[i].type = (char \*) calloc(1, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

ptr2[i].type = "c";

}

}

}

void output2(int n, struct student\_add \*ptr2) //вывод информации о родственниках студентов в зависимости от запроса

{

int i;

char \*q;

printf("\n\nЕсли вы хотите получить информацию о родственниках студентов полных семей, введите а; о родственниках студентов многодетных семей, введите b; о родственниках студентов неполных семей, введите c: ");

if((q = (char \*) calloc(1, sizeof(char))) == NULL)

{

printf("\nПамять не выделена");

return;

}

fflush(stdin);

fgets(q, 2, stdin);

printf("\nИнформация о родственниках студентов: ");

printf("\nНомер ФИО");

for(i=0; i<n; i++)

{

switch(q[0])

{

case 'a': if(strcmp(ptr2[i].type, q) == 0) printf("\n%-3d%s%s%s", i+1, ptr2[i].rel.a.father, ptr2[i].rel.a.mother, ptr2[i].rel.a.brother); break;

case 'b': if(strcmp(ptr2[i].type, q) == 0) printf("\n%-3d%s%s%s%s", i+1, ptr2[i].rel.b.father, ptr2[i].rel.b.mother, ptr2[i].rel.b.brother, ptr2[i].rel.c.sister); break;

case 'c': if(strcmp(ptr2[i].type, q) == 0) printf("\n%-3d%s%s%s", i+1, ptr2[i].rel.c.mother, ptr2[i].rel.c.brother, ptr2[i].rel.c.sister); break;

}

}

for(i=0; i<n; i++) //освобождение памяти, выделенной под указатели в структуре с объединением

{

free(ptr2[i].q1);

free(ptr2[i].q2);

free(ptr2[i].type);

}

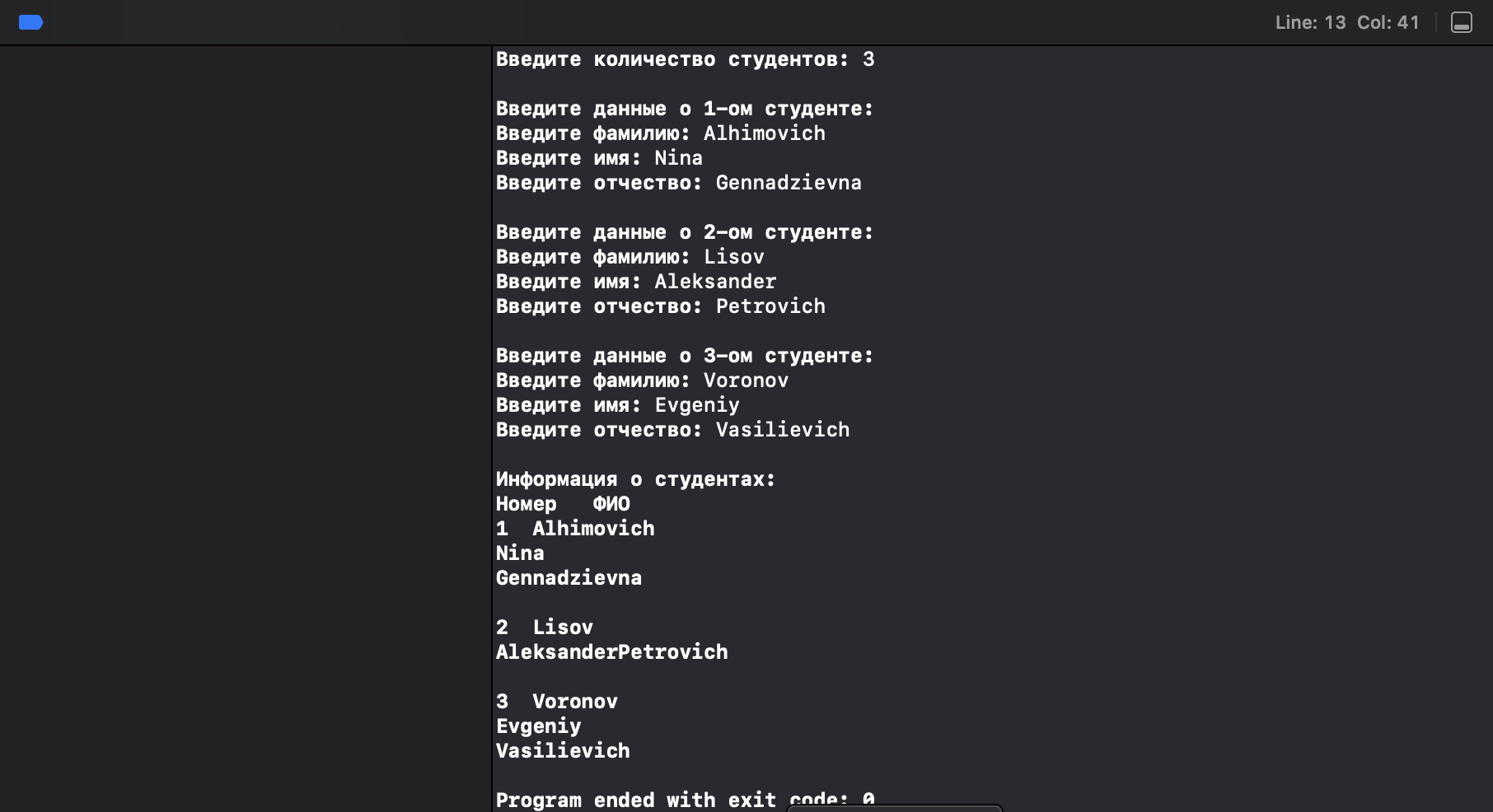
free(ptr2); //освобождение памяти, выделенной под указатель на структуру с объединением

free(q); //освобождение памяти, выделенной под указатель на переменную-индикатор

}

Примеры работы программы:

В случае, если признаки требуют работу с простой структурой:



В случае, если признаки требуют работу со структурой с объединением: