Міністерство освіти та науки України

Інститут «Телекомунікації, радіоелектроніки та радіотехніки»

Звіт до лабораторної роботи № 18

З дисципліни «Програмування частина 2»

Структури та об’єднання даних

Виконав студент групи АП-12:

Дроф’як Святослав

Перевірив викладач:

Пелех Н.В.

Львів 2024

Мета роботи: ознайомитися з поняттями структури та об’єднання даних , навчитися їх використовувати у процесі програмування.

# Хід роботи:

1. Ознайомитися з теоретичними відомостями.
2. Здійснити виконання прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати виконання у звіті.
3. Написати програму для виведення нижчепредставленої інформації шляхом використання структури. Ім’я, вага, висота, вік – вводяться з клавіатури ( вказати довільні дані).

Інформація про працівника

Ім’я

Вага

Висота

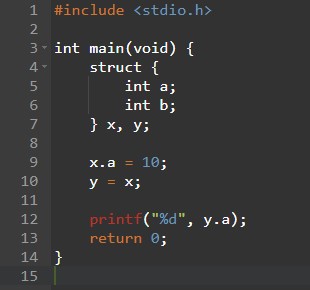
Вік

1. Оформити звіт.

**Завдання 1**

**Реалізація:**

Програмна реалізація мовою С



Код завдання 1, приклад 1

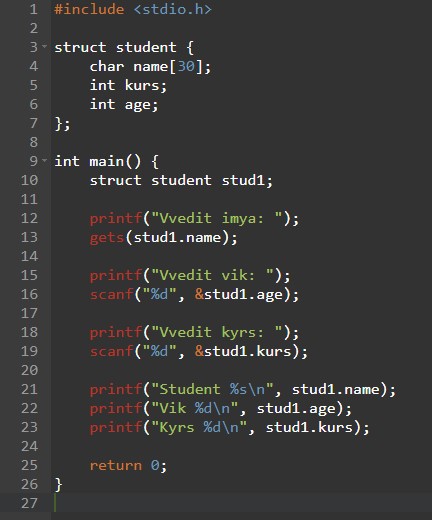


Скріншот виконання коду, приклад 1

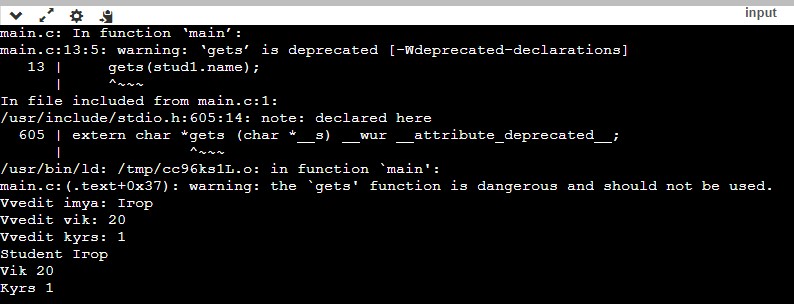
Ця програма на мові програмування C демонструє використання структур і присвоєння значень. Ось що відбувається крок за кроком:

1. Включається стандартна бібліотека вводу/виводу stdio.h, яка необхідна для використання функції printf.
2. Оголошується функція main, яка є точкою входу в програму.
3. Всередині main оголошується структура без імені, яка містить два цілочисельні поля: a та b.
4. Створюються дві змінні цієї структури: x та y.
5. Змінній a структури x присвоюється значення 10.
6. Структура y отримує всі значення з x завдяки присвоєнню структур.
7. Виводиться значення поля a структури y, яке є 10.
8. Програма завершується з кодом повернення 0, що означає успішне виконання.

Отже, коли ви запустите цю програму, вона виведе на екран число 10.



Код завдання 1, приклад 2

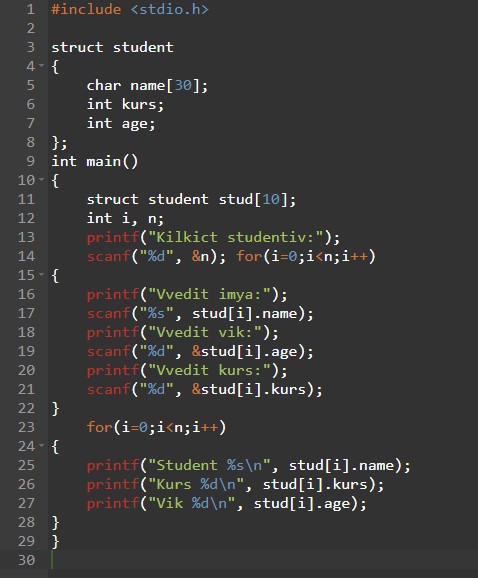


Скріншот виконання коду, приклад 2

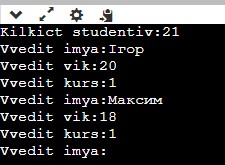
Ця програма на мові C створює і використовує структуру для зберігання інформації про студента. Ось короткий опис дій програми:

1. Підключається заголовочний файл stdio.h для використання функцій вводу/виводу.
2. Оголошується структура student, яка містить рядок для імені, ціле число для курсу та віку.
3. У функції main створюється змінна stud1 типу student.
4. Програма запитує у користувача ім’я, вік та курс, використовуючи printf для виводу повідомлення та gets та scanf для зчитування введення.
5. Введені дані зберігаються у полях структури stud1.
6. Програма виводить збережену інформацію про студента за допомогою printf.

**Важливо**: Функція gets є небезпечною, оскільки вона не перевіряє розмір буфера і може призвести до переповнення буфера. Рекомендується використовувати fgets замість gets.



Код завдання 1, приклад 3



Скріншот виконання коду, приклад 3

Ця програма на мові C використовує масив структур для зберігання інформації про декількох студентів. Ось короткий опис її дій:

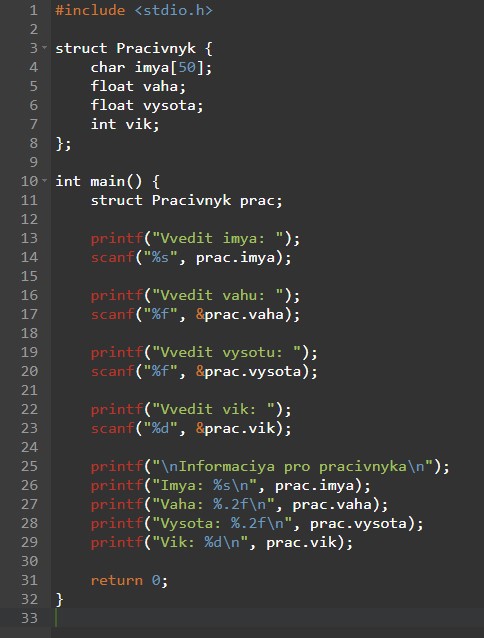
1. Підключається заголовочний файл stdio.h.
2. Оголошується структура student з полями для імені, курсу та віку.
3. У main створюється масив stud з 10 елементів типу student.
4. Користувачу пропонується ввести кількість студентів n.
5. У циклі for користувач вводить дані для кожного студента: ім’я, вік, курс.
6. Другим циклом for програма виводить інформацію про кожного студента.

Програма зчитує дані через консоль і виводить їх, використовуючи стандартні функції вводу/виводу C. Зверніть увагу, що використання scanf для рядків може бути небезпечним без обмеження довжини вводу, щоб уникнути переповнення буфера.

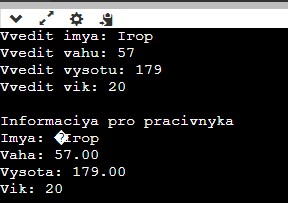
**Завдання 2**

**Реалізація:**

Програмна реалізація мовою С



Код завдання 2



Скріншот виконання коду, завдання 2

Ця програма на мові C створює структуру для зберігання даних про працівника і зчитує ці дані з консолі. Ось що вона робить:

1. Підключається заголовочний файл stdio.h для функцій вводу/виводу.
2. Оголошується структура Pracivnyk з полями для імені, ваги, висоти та віку.
3. У функції main створюється змінна prac типу Pracivnyk.
4. За допомогою printf та scanf програма запитує у користувача ім’я, вагу, висоту та вік працівника.
5. Введені дані зберігаються у відповідних полях структури prac.
6. Програма виводить зібрану інформацію про працівника.

**Зауваження**: Використання %s у scanf без обмеження довжини може призвести до переповнення буфера. Рекомендується використовувати %49s для зчитування рядка, щоб уникнути цієї проблеми. Також, для більшої безпеки, краще використовувати fgets замість scanf для зчитування рядків.

# Контрольні питання

1. Дайте визначення поняття «структура».

**Визначення поняття “структура”**: В мові програмування C структура (або “структура даних”) представляє собою **составний тип даних**, який може містити **різні компоненти** (поля) різних типів. Структура дозволяє групувати ці компоненти під одним ім’ям для зручного доступу.

2. Яким чином здійснюється оголошення структури?

**Оголошення структури**: Для оголошення структури використовується ключове слово struct. Синтаксис оголошення структури виглядає так:

struct ім'я\_структури {

тип компоненти1;

тип компоненти2;

// ...

};

struct person {

int age;

char \*name;

};

3. Охарактеризуйте синтаксис об’єднання даних.

**Синтаксис об’єднання даних**: Об’єднання (union) в C дозволяють визначити свій власний тип даних, який також містить набір елементів. Однак, на відміну від структури, всі елементи об’єднання мають **нульове зміщення**. Це означає, що різні елементи об’єднання займають **один і той же регіон пам’яті**. Синтаксис оголошення об’єднання схожий на оголошення структури:

union ім'я\_об'єднання {

тип елемент1;

тип елемент2;

// ...

};

union ascii {

int digit;

char letter;

};

4. Які операції не можна застосовувати до структур?

**Операції, які не можна застосовувати до структур**:

* Структури не можна порівнювати між собою (наприклад, struct1 == struct2).
* Структури не можна інкрементувати або декрементувати (наприклад, struct1++).
* Структури не можна використовувати в якості операндів логічних операцій (наприклад, struct1 && struct2).

**Висновок:**

Ця лабораторна робота надала мені можливість детальніше ознайомитися з поняттями структур та об'єднань даних у мові програмування C. Під час виконання роботи я зрозумів, як створювати та використовувати структури для організації та зберігання різноманітних типів даних.

Написання програми для зберігання інформації про працівника за допомогою структури дало мені можливість практично застосувати отримані знання. Введення даних про працівника з клавіатури та їх подальше виведення на екран стало доступним завдяки використанню структури.

Таким чином, завдяки цій лабораторній роботі я отримав навички використання структур у програмуванні на мові C та зрозумів їх важливість у створенні більш структурованих програм.