Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

**ЗВІТ**

виконання лабораторної роботи №3

**ТЕМА:** Мова програмування сі, компіляція, функції вводу/виводу.

**МЕТА:** Вивчити поняття: компіляції, функцій вводу/виводу.

**СПИСОК ВИКОРИСТАНОГО ПЗ:** онлайн компілятор мови програмування Сі.

**ВИКОНАВ:**

Студент 1-го курсу

Групи ФЕП-14

Іванський Остап Юрійович

**ПЕРЕВІРИВ:**

Асистент

Вдовиченко Віктор Мар’янович

Львів 2024

**ЗМІСТ**

1. Теоретичні відомості.
2. Програма яка ілюструє роботу функції printf().
3. Програма яка реалізує введення кількох стрічок символів та виведення їх у відформатованому вигляді.
4. Висновок.

**ХІД РОБОТИ**

**1.Теоретичні відомості.**

Мова Сі була використана для створення широко поширених операційних систем - Linux, Windows, OS X, iOS і Android, оскільки мова надає вільний доступ до апаратних ресурсів комп'ютера. Порівнюючи їїз іншими мовами програмування, такими як Python або Matlab, необхідно відзначити, що в мові Сі відсутня вбудована реалізація таких розвинених мовних засобів, як високорівневі функції для роботи з файлами, порівняння з шаблоном, матричні операції і взаємодія з графічним інтерфейсом користувача. Також в ній відсутні ресурси, що дозволяють виявити помилки при зверненні до пам'яті, помилки виходу за межі масиву. З іншого боку це є перевагою мови Сі котра дозволяє хакерам проникати в комп'ютерні системи, де використовується програмне забезпечення, розроблене без належної уваги до безпеки.

Зазвичай, програма на мові Сі містить одну або кілька функцій. Кожна програма повинна мати функцію з ім'ям main, яка служить стартовою точкою для виконання програми і в кожній окремій програмі ця функція є єдина. Крім цієї головної функції, більшість програм на мові Сі містять набір функцій, які знаходяться в тексті програми і / 2 або в бібліотеках. Наш приклад hello.c складається з директиви включення в програму заголовного файлу stdio.h, оголошення функції main і її реалізації. Для підключення до програми стандартної бібліотеки використовується ключове слово #include, після якого вказується назва заголовного файлу даної бібліотеки, взята в трикутні дужки. Заголовок: #include Заголовки складаються з оголошень функцій, потрібних програмі. У нашому випадку, програма використовує функцію printf, яка знаходиться в стандартній бібліотеці введення / виведення і оголошена в заголовному файлі stdio.h. Робота програми починається з виконання операторів, що містяться в функції з ім'ям main. Набір цих операторів називається тілом функції main. Будь-яка програма на Сі повинна мати одну і тільки одну функцію main. Тіло функції містить послідовність операторів, кожний з яких має закінчуватися крапкою з комою. Ключове слово int вказує, що результат функції (або значення, що повертається) має цілочисельний тип. Значення, що повертається показує, чи була програма виконана успішно (якщо так, то функція main повертає нуль, якщо програма завершилася аварійно, то функція main повертає код помилки). По завершенню програми, результат main передається в те оточення, з якого вона була запущена. Тіло функції: printf("Hello World!\n"); Тіло функції main з нашого прикладу містить єдиний виклик функції printf, що друкує рядок "Hello world!", яка закінчується символом переведення рядка "\n". У загальних рисах, всі програми влаштовані подібно до нашого прикладу hello.c, з тією різницею, що складні програми можуть складатися з мільйонів рядків тексту, розбитих на безліч функцій і файлів.

**Запуск Сі-програми**.

Програми на мові Сі можуть виконуватися процесорами з різними системами команд. Переносимість програм - це ще одна перевага мови програмування Сі. Перед запуском програма повинна бути скомпільована Сі-компілятором. Існує кілька відмінних реалізацій Сі-компіляторів, включаючи cc (C компілятор) і gcc (GNU C компілятор). Компілятор gcc доступний для вільного використання. Він поставляється в складі всіх дистрибутивів операційної системи Linux і не вимагає додаткових дій по інсталяції. В операційній системі Windows може знадобитися установка пакета програм Cygwin. Процес підготовки файлу з програмою, його компіляція і виконання коду програми однакові для всіх програм на мові Сі.

**Функції вводу/виводу, printf та scanf з бібліотеки stdio printf** - вивід форматованого рядка в стандартний потік виведення.

**синтаксис**: #include int printf (const char \* format, ...);

**аргументи**: format - вказівник на рядок з описом формату.

**Значення, що повертається:**

При успішному завершенні виведення повертається кількість виведених символів. При помилці повертається негативне число.

**Опис:**

Функція printf виводить в стандартний потік виведення рядок, відформатований відповідно до правил, зазначених в рядку, на які вказує аргумент format. Правила задаються набором трьох типів директив:

1. Звичайні символи, які виводяться без зміни (крім '%' і '\');

2. Специфікатори формату;

3. Спеціальні символи. Специфікатори формату.

Кожен специфікатор формату починається з символу '%' і має такий вигляд:

% [прапорці] [ширина] [точність] [модифікатори] [тип перетворення]

Специфікатор [прапорець] можна не вказувати.

**Ширина**

Специфікатор [ширина] задає мінімальний розмір виведеного числа в символах. Якщо кількість символів в виведеному числі менша зазначеної мінімальної ширини, то символи, котрих бракує заповнюється нулями або пропусками зліва чи справа в залежності від зазначених прапорців. Ширина вказується або цілим числом, або символом \* з подальшим зазначенням імені змінної типу int, що містить значення ширини, перед аргументом до якого він відноситься. Якщо аргумент має від'ємне значення, то він еквівалентний відповідному позитивному значенню з прапорцем "-".

Специфікатор [ширина] можна не вказувати.

**Точність**

Дія специфікатора [точність] залежить від типу виведеного числа:

- Для типів d, i, o, u, x, X визначає мінімальне число виведених цифр. Якщо кількість виведених цифр в числі менше, ніж зазначено в специфікатором [точність], то число, що виводиться буде доповнено нулями зліва. Наприклад, якщо при виведенні числа 126 вказати точність 4, то на екран буде виведено число 0126

- Для типів a, A, e, E, f, F визначає кількість виведених цифр після коми. Якщо в виведеному числі кількість значущих цифр після коми менше зазначеної точності, то відсутні символи виводяться нулями праворуч від числа. Якщо більше, то зайві цифри не виводяться. Наприклад, якщо при виведенні числа 126.345 вказати точність 2, буде виведено на екран число 126.34, а якщо вказати точність 5, то на екран буде виведено число 126.34500.

- Для типів g і G визначає максимальне виведене число цифр. Наприклад, якщо при виведенні числа 126.345 вказати точність 4, буде виведено на екран число 126.3. Якщо при виведенні числа 1242679.23 вказати точність 3, буде виведено на екран число 1.24е + 06.

Точність вказується у вигляді символу точка, за яким слідує десяткове число або символ "\*", з подальшою вказівкою імені змінної типу int, що містить значення точності, перед аргументом до якого він відноситься.

Специфікатор [точність] можна не вказувати.

**Модифікатор**

Специфікатор [модифікатор] визначає розмір даних, що виводяться (char, short, long, longlong). Специфікатори використовуються для виведення чисел типу: char, short int, long int, long long int, long double або для явного перетворення даних, що виводяться. Наприклад, якщо є змінна типу int, а необхідно вивести її як short int.

Специфікатор [модифікатор] можна не вказувати.

**Тип перетворення**

Специфікатор [тип перетворення] визначає як треба інтерпретувати і виводити число, наприклад як знакове цілочислене в десятковому вигляді, або беззнакове цілочислене в шістнадцятковому, або як число з плаваючою комою і так далі.

**Форматований ввід - scanf**

Функція scanf є аналогічною до printf, але тільки для вводу. Вона надає багато з тих самих можливостей перетворення, щоправда, в зворотному напрямку.

int scanf (char \* format, arg1, arg2,...);

scanf зчитує знаки зi стандартного вводу, інтерпретуючи їх відповідно до специфiкацiї, вказаної форматом (format), i зберігає результат за допомогою решти аргументів. Аргумент формату описано нижче; решта аргументів, кожен з яких повинен бути вказівником, вказує на те, де відповідний перетворений ввід потрібно зберегти.

scanf зупиняється, якщо вона вичерпає свій список формату, або коли ввід не збігається із контрольною специфікацією. Вона повертає як значення число елементів вводу, які були сприйняті i яких було присвоєно. Це можна використати, щоб дізнатися, скільки об’єктів було введено.

Керуюча стрічка, на яку вказує format, складається з символів трьох типів:

• Специфікатори формату

• Спеціальні символи

• Інші символи (не спеціальні)

Специфікатори формату слідують за символом % (процент) и вказують scanf() тип даних, які будуть зчитані наступними. Ці параметри є аналогічними до тих, що використовує printf().

Наприклад, %s зчитує стрічку, a %d зчитує десяткову змінну цілого типу.

Стрічка формату зчитується зліва направо, при цьому встановлюється відповідність між кодами формату і аргументами зі списку аргументів.

Всі змінні, які приймають значення від функції scanf(), повинні знаходитися за їхніми адресами. Це означає, що всі аргументи функції повинні бути вказівниками на змінні. Таким чином, Сі створює можливість передачі по посиланню, і це дозволяє функції змінювати вміст аргументу.

Наприклад, для того, щоб зчитати ціле число і присвоїти його значення змінній count, необхідно скористатися наступним зверненням до scanf():

scanf("%d", &count);

Стрічки є масивами символів, і ім’я масиву є адресою першого елемента масиву, тобто по суті є вказівником. Тому для зчитування стрічки в масив символів address, можна використати команду

scanf("%s", address);

В цьому випадку ім’я address уже є вказівником, і не потребує префіксу &. Елементи даних вводу повинні розділятися пробілами, знаками табуляції або нового рядка. Розділові знаки (крапки, коми і т.д.) не вважаються розділювачами. Це означає, що оператором

scanf("%d%d", &r, &с);

послідовність 10 20 буде сприйнята, а послідовність 10,20 – ні.

Комбінація %\*, розміщена перед специфікатором формату, зчитує дані вказаного типу, але блокує їхнє присвоювання. Т.ч., код

scanf ("%d%\*c%d", &х, &у);

при вводі послідовності 10/20 присвоює значення 10 змінній х, відкидає символ / і присвоює значення 20 змінній у.

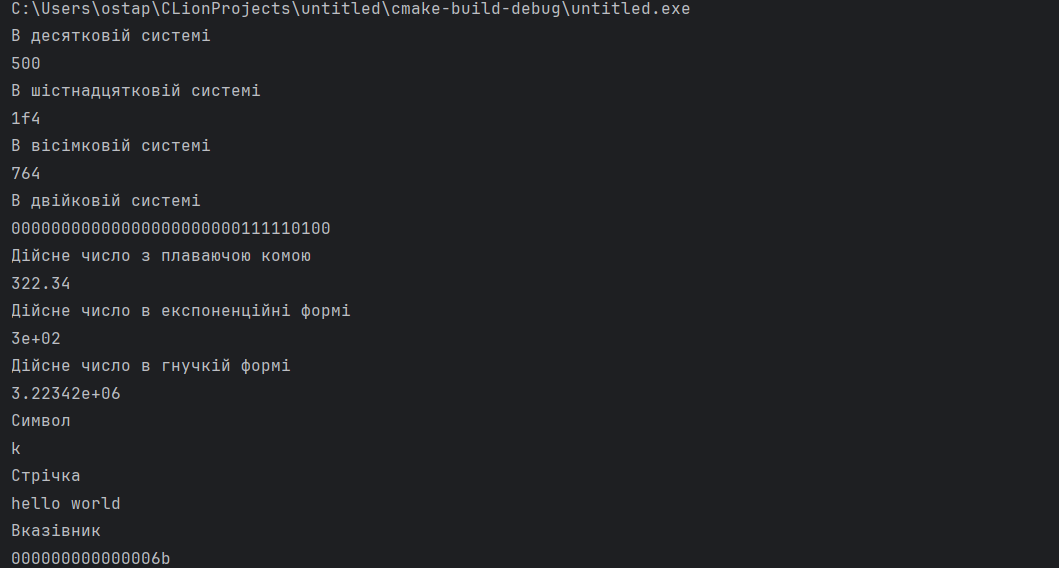
Модифікатор максимальної ширини поля є цілим числом, яке розміщується між знаком % і специфікатором формату. Він обмежує кількість зчитуваних символів для довільного поля. Наприклад, якщо необхідно зчитати не більше 20 символів в масив address, слід написати

scanf ("%20s", address);

**2.Програма яка ілюструє роботу функції printf().**

#include <stdio.h>  
  
void tobin(int a) {  
 for (int i = 31; i >= 0; i--) {  
 int bit = (a >> i) & 1;  
 printf("%d", bit);  
 }  
 printf("\n");  
}  
  
  
int main(void) {  
 int value = 500;  
 double v2 = 322.34;  
 double v3 = 3223422.34;  
 char ch = 'k';  
 char str[]="hello world";  
  
 printf("В десятковій системі\n");  
 printf("%d\n", value);  
  
 printf("В шістнадцятковій системі\n");  
 printf("%x\n", value);  
  
 printf("В вісімковій системі\n");  
 printf("%o\n", value);  
  
 printf("В двійковій системі\n");  
 tobin(value);  
  
 printf("Дійсне число з плаваючою комою\n");  
 printf("%.2f \n",v2);  
  
 printf("Дійсне число в експоненційні формі \n");  
 printf("%.e \n",v2);  
  
 printf("Дійсне число в гнучкій формі \n");  
 printf("%g \n",v3);  
  
 printf("Символ \n");  
 printf("%c \n",ch);  
  
 printf("Стрічка \n");  
 printf("%s \n",str);  
  
 printf("Вказівник \n");  
 printf("%p \n",ch);  
}

Результат:



**3.Програма яка реалізує введення кількох стрічок символів та виведення їх у відформатованому вигляді.**

#include <stdio.h>  
  
#define MAX\_STUDENTS 100  
#define NAME\_LENGTH 50  
#define EMAIL\_LENGTH 100  
#define COLOR\_LENGTH 30  
  
typedef struct {  
 char name[NAME\_LENGTH];  
 char email[EMAIL\_LENGTH];  
 char color[COLOR\_LENGTH];  
} Student;  
  
void printTableHeader() {  
 printf("| %-7s | %-30s | %-40s | %-18s |\n", "№", "Surname, Initials",  
 "Email", "Color");  
 printf("|-------|--------------------------------|---------------------------"  
 "---------------|--------------------|\n");  
}  
  
void printStudent(Student student, int index) {  
 printf("| %-5d | %-30s | %-40s | %-18s |\n", index + 1, student.name,  
 student.email, student.color);  
}  
  
int main() {  
 int n;  
 Student students[MAX\_STUDENTS];  
  
 printf("Enter count of students (max %d): ", MAX\_STUDENTS);  
 scanf("%d", &n);  
 getchar(); // очищення буфера після scanf  
  
 if (n > MAX\_STUDENTS || n <= 0) {  
 printf("Wrong number.\n");  
 return 1;  
 }  
  
 for (int i = 0; i < n; i++) {  
 printf("Enter surname, initials %d: ", i + 1);  
 scanf(" %49[^\n]", students[i].name);  
  
 printf("Enter email %d: ", i + 1);  
 scanf(" %99[^\n]", students[i].email);  
  
 printf("Enter favorite color %d: ", i + 1);  
 scanf(" %29[^\n]", students[i].color);  
 }  
  
 // Виведення таблиці  
 printTableHeader();  
 for (int i = 0; i < n; i++) {  
 printStudent(students[i], i);  
 }  
 printf("|-------|--------------------------------|---------------------------"  
 "---------------|--------------------|\n");  
  
 return 0;  
}

**4.Висновок**

Під час роботи над завданням вийшло попрацювати з функціями вводу та виводу даних, методами форматування та запису стрічки. Було корисно зробити і зрозуміти принцип роботи функції для виводу числа у двійковому коді.