Лаб 1 Соловйов ІТ-32-2

Завдання:

1. Поділити два цілих числа 5 та 2 і вивести результат як float. (0.2 бала)
2. Оголосити константну змінну типу float та присвоїти їй значення 3.14. Вивести її значення на екран. (0.2 бала)
3. Написати програму, яка оголошує змінну типу char та виводить її ASCII-код. (0.2 бала)
4. Користувач вводить ціле і дробове число. Виведіть їх суму. (0.2 бала)
5. Напишіть програму, яка зчитує символ та виводить наступний символ у таблиці ASCII.(0.2 бала)

Код:

#define \_CRT\_SECURE\_NO\_WARNINGS // Щоб Visual Studio не ругався на scanf

#include <stdio.h>

#include <stdlib.h>

int main() {

system("chcp 1251");

int choice;

int ShowMenu = 1;

do {

if (ShowMenu == 1)

{

printf("\nМеню завдань:\n");

printf("1. Поділити два цілих числа 5 та 2 і вивести результат як float\n");

printf("2. Оголосити константну змінну типу float та присвоїти їй значення 3.14\n");

printf("3. Оголосити змінну типу char та вивести її ASCII-код\n");

printf("4. Ввести ціле і дробове число та вивести їх суму\n");

printf("5. Зчитати символ і вивести наступний символ у таблиці ASCII\n");

printf("-1. Вимкнути/увімкнути показ меню\n");

printf("0. Вийти\n");

}

printf("Оберіть номер завдання: ");

scanf("%d", &choice);

switch (choice) {

case 1: {

int a = 5, b = 2;

float result = (float)a / b;

printf("Результат: %.2f\n", result);

break;

}

case 2: {

const float pi = 3.14f;

printf("Константа: %.2f\n", pi);

break;

}

case 3: {

char ch = 'A'; // або можна зчитувати від користувача

printf("Символ: %c, ASCII-код: %d\n", ch, ch);

break;

}

case 4: {

int intVal;

float floatVal;

printf("Введіть ціле число: ");

scanf("%d", &intVal);

printf("Введіть дробове число: ");

scanf("%f", &floatVal);

float sum = intVal + floatVal;

printf("Сума: %.2f\n", sum);

break;

}

case 5: {

char inputChar;

printf("Введіть символ: ");

scanf(" %c", &inputChar);

char nextChar = inputChar + 1;

printf("Наступний символ: %c\n", nextChar);

break;

}

case -1:

ShowMenu = !ShowMenu;

break;

case 0:

printf("Вихід з програми.\n");

break;

default:

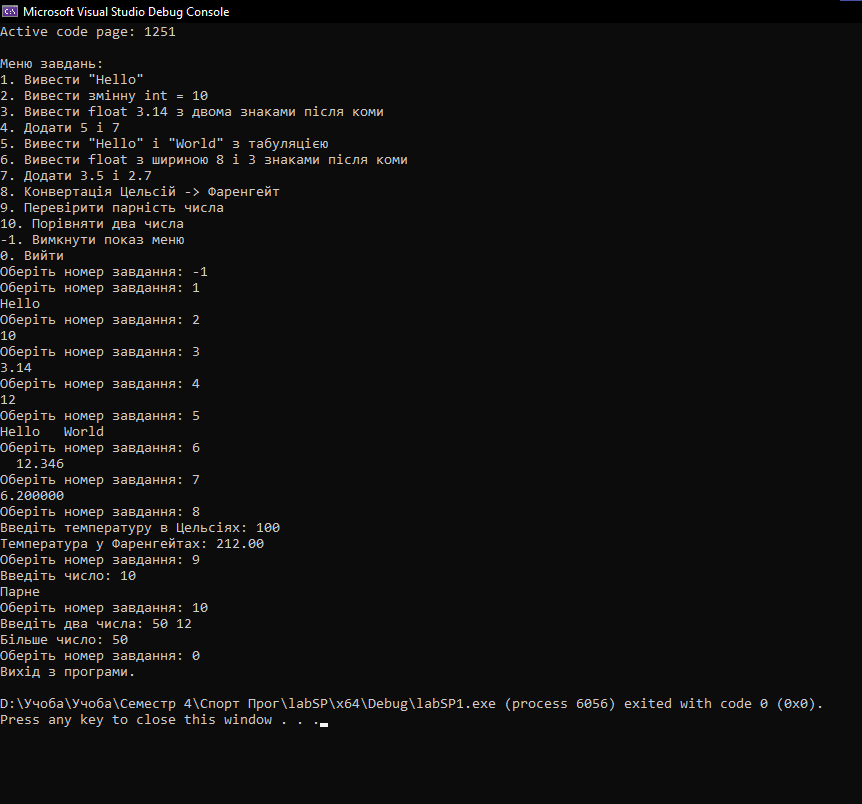
printf("Невірний вибір. Спробуйте ще раз.\n");

}

} while (choice != 0);

return 0;

}

Результат роботи:

Github: https://github.com/SolDima/labSP/tree/main/lab1

Творче завдання 1:

Що буде виведено на екран в результаті виконання наступного коду на мові C?

#include <stdio.h>

int main() {

char c = 'A';

printf("**%d**\n", c + 1);

return 0;

}

Варіанти відповідей:

A) 'B'  
B) A1  
C) 66  
D) B1

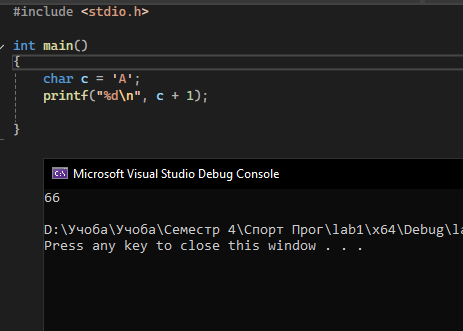
Правильна відповідь: C) 66

A) 'B' — було б правильним, якби ми написали printf("**%c**\n", c + 1);

B) A1

Правильна відповідь — C)  
Символ 'A' у таблиці ASCII має код 65. У виразі c + 1 відбувається додавання 65 + 1, так як ми виводим %d, то виводиться 66

D) B1



Творче завдання 2:

У якості другого творчого завдання я спробував розібратись з побітовим зсувом і використати його для ділення і множення значень.

case 7: { // Творче завдання 2

unsigned int num;

printf("Введіть невід'ємне ціле число: ");

scanf("%u", &num);

unsigned int multiplied = num << 1;

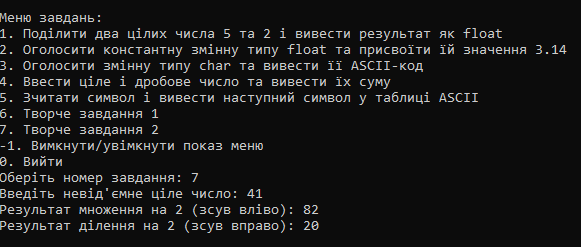
unsigned int divided = num >> 1;

printf("Результат множення на 2 (зсув вліво): %u\n", multiplied);

printf("Результат ділення на 2 (зсув вправо): %u\n", divided);

break;

}



Побітовий зсув вліво (<<)

Зсуває всі біти числа вліво на вказану кількість позицій. Кожен зсув вліво на 1 множить число на 2.

Приклад:

5 (в двійковій системі): 00000101

5 << 1 → 00001010 = 10

Побітовий зсув вправо (>>)

Зсуває біти вправо. Кожен зсув вправо на 1 ділить число на 2 (без залишку для цілих чисел).

Приклад:

5 (в двійковій системі): 00000101

5 >> 1 → 00000010 = 2

Як я зрозумів, до від'ємних чисел не рекомендуєтсья застосовувати такий метод ділення або множення, адже це може спрацювати неправильно.

Також варто враховувати, що у ситуації ділення, округлення відбувається у меншу сторону.