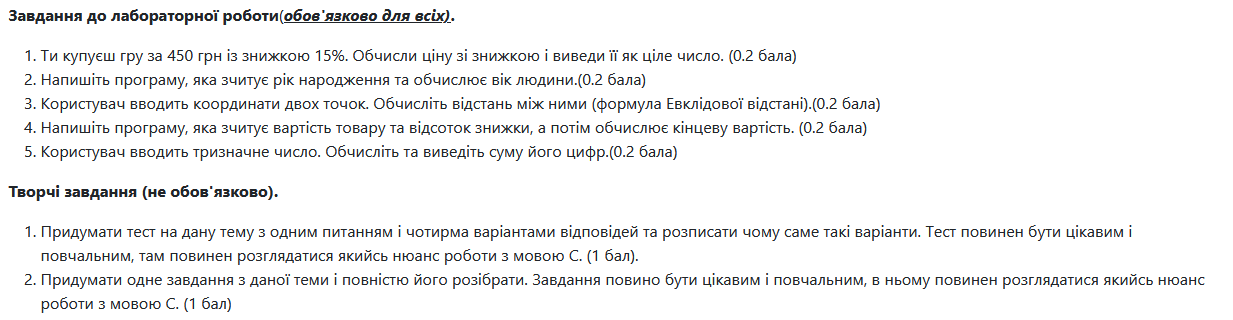
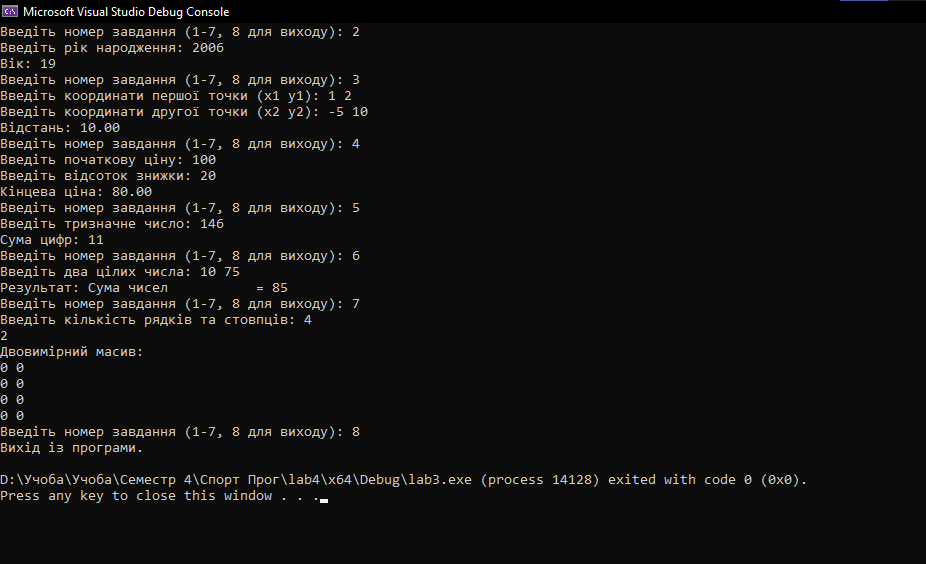
Соловйов лаб 4 ІТ-32-2

Завдання





Код:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#define FORMAT\_OUTPUT "Результат: %-20s = %d\n"

int main() {

system("chcp 1251 > nul");

int choice;

do {

printf("Введіть номер завдання (1-7, 8 для виходу): ");

scanf("%d", &choice);

switch (choice) {

case 1: {

float price = 450.0;

float discount = 0.15;

int final\_price = (int)(price \* (1 - discount));

printf("Кінцева ціна після знижки 15%: %d\n", final\_price);

break;

}

case 2: {

int birth\_year, age;

printf("Введіть рік народження: ");

scanf("%d", &birth\_year);

age = 2025 - birth\_year;

printf("Вік: %d\n", age);

break;

}

case 3: {

float x1, y1, x2, y2;

printf("Введіть координати першої точки (x1 y1): ");

scanf("%f %f", &x1, &y1);

printf("Введіть координати другої точки (x2 y2): ");

scanf("%f %f", &x2, &y2);

float distance = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

printf("Відстань: %.2f\n", distance);

break;

}

case 4: {

float price, discount\_percent;

printf("Введіть початкову ціну: ");

scanf("%f", &price);

printf("Введіть відсоток знижки: ");

scanf("%f", &discount\_percent);

float final\_price = price \* (1 - discount\_percent / 100);

printf("Кінцева ціна: %.2f\n", final\_price);

break;

}

case 5: {

int number;

printf("Введіть тризначне число: ");

scanf("%d", &number);

if (number >= 100 && number <= 999) {

int sum = (number / 100) + ((number / 10) % 10) + (number % 10);

printf("Сума цифр: %d\n", sum);

}

else {

printf("Будь ласка, введіть тризначне число.\n");

}

break;

}

case 6: { //Творче завдання 1

int a, b, sum;

printf("Введіть два цілих числа: ");

scanf("%d %d", &a, &b);

sum = a + b;

printf(FORMAT\_OUTPUT, "Сума чисел", sum);

break;

}

case 7: { //Творче завдання 2

int rows, cols;

printf("Введіть кількість рядків та стовпців: ");

scanf("%d %d", &rows, &cols);

int\*\* array = (int\*\*)calloc(rows, sizeof(int\*));

for (int i = 0; i < rows; i++) {

array[i] = (int\*)calloc(cols, sizeof(int));

}

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

printf("%d ", array[i][j]);

}

printf("\n");

}

for (int i = 0; i < rows; i++) {

free(array[i]);

}

free(array);

break;

}

case 8:

printf("Вихід із програми.\n");

break;

default:

printf("Невірний вибір.\n");

}

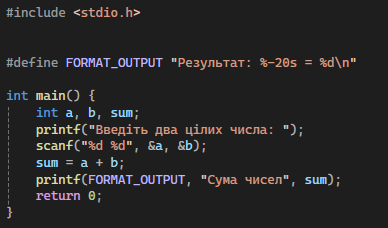
} while (choice != 8);

return 0;

}

Творче завдання 1

Що виведе програма?



Що виведе програма в шостому кейсі, якщо користувач введе два числа: 5 і 3?

Варіанти відповіді:

Результат: Сума чисел = 8

Сума чисел: 8

Результат: 5 + 3 = 8

Помилку

Правильна відповідь: 1. Результат: Сума чисел = 8

Творче завдання 2

У цьому творчому завдання я хочу розказати про динамічне виділення пам’яті у Сі на прикладі динамічних двовимірних масивів.

Динамічні масиви в Сі — це масиви, розмір яких визначається під час виконання програми, а не під час компіляції. Вони створюються за допомогою функцій malloc або calloc з бібліотеки <stdlib.h>, які виділяють для них пам’ять. malloc виділяє неініціалізовану пам’ять, тоді як calloc ініціалізує її нулями. Після використання пам’ять потрібно звільнити за допомогою free, щоб уникнути витоків пам’яті. Для двовимірного масиву потрібно виділяти пам’ять для масиву вказівників і для кожного рядка окремо.

Приклад створення динамічного масиву та очищення пам’яті

int rows, cols;

printf("Введіть кількість рядків та стовпців: ");

scanf("%d %d", &rows, &cols);

int\*\* array = (int\*\*)calloc(rows, sizeof(int\*));

for (int i = 0; i < rows; i++) {

array[i] = (int\*)calloc(cols, sizeof(int));

}

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

printf("%d ", array[i][j]);

}

printf("\n");

}

for (int i = 0; i < rows; i++) {

free(array[i]);

}

free(array);

break;

github: https://github.com/SolDima/labSP