Лекція 7.
Цикл з післяумовою do … while.
Операція «кома»

Мова Сі дозволяє реалізовувати цикл з **постумовою (післяумовою)** за допомогою оператора **do…while**:

```
do
    оператор;
while ( умова ) ;
```

тіло циклу

буде повторюватися, доти, доки умова буде істинною

Особливість циклу:

- спочатку виконується тіло циклу, а потім перевіряється умова;

- тіло циклу виконається мінімум один раз (навіть якщо умова одразу буде хибною);

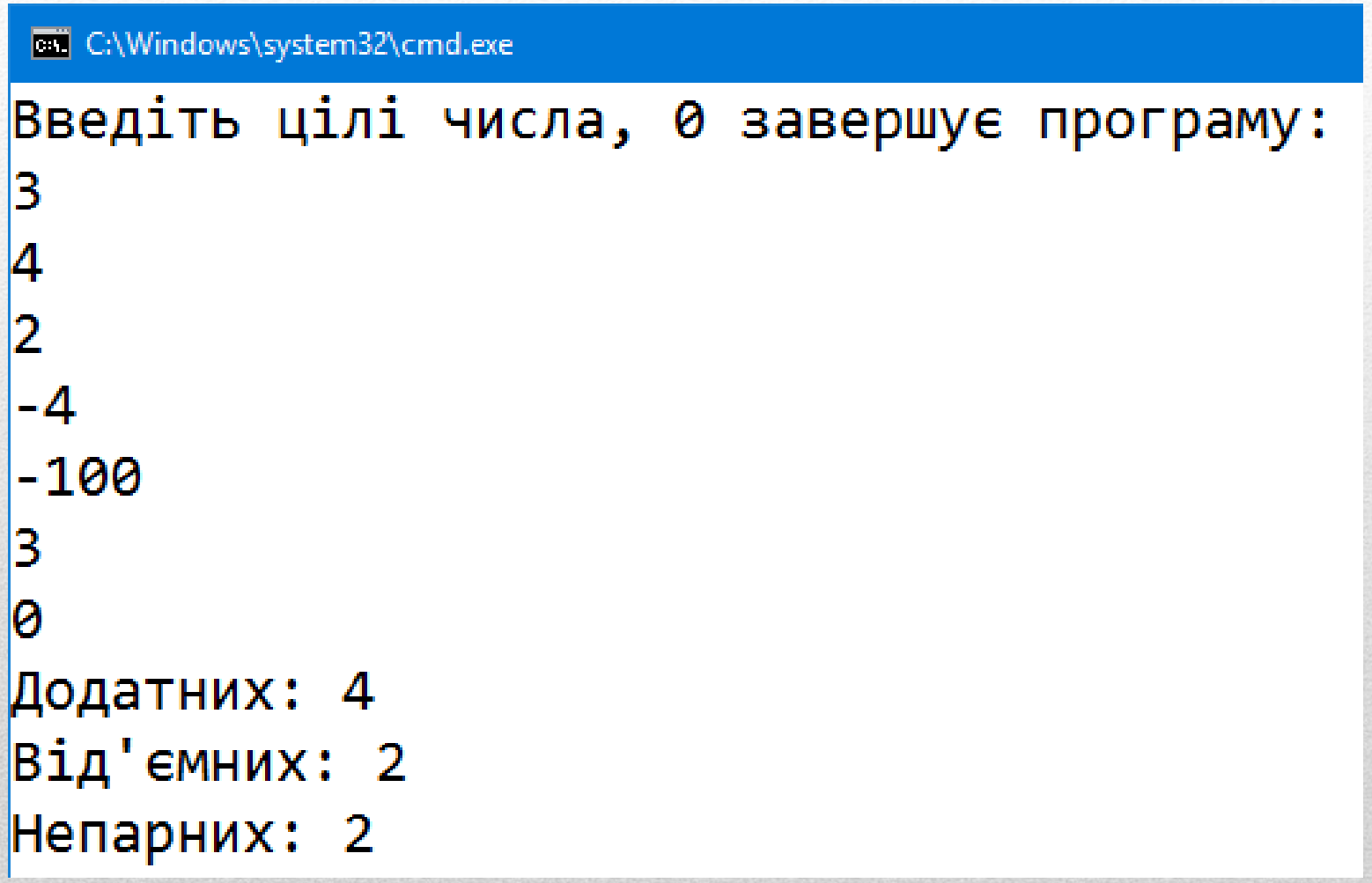
Коли зручно використовувати цикл **do … while ?**

Завдання. Користувач вводить послідовно один за одним цілі числа, закінчуючи введення нулем. Порахувати у введеній послідовності кількість непарних, додатних та від’ємних чисел.

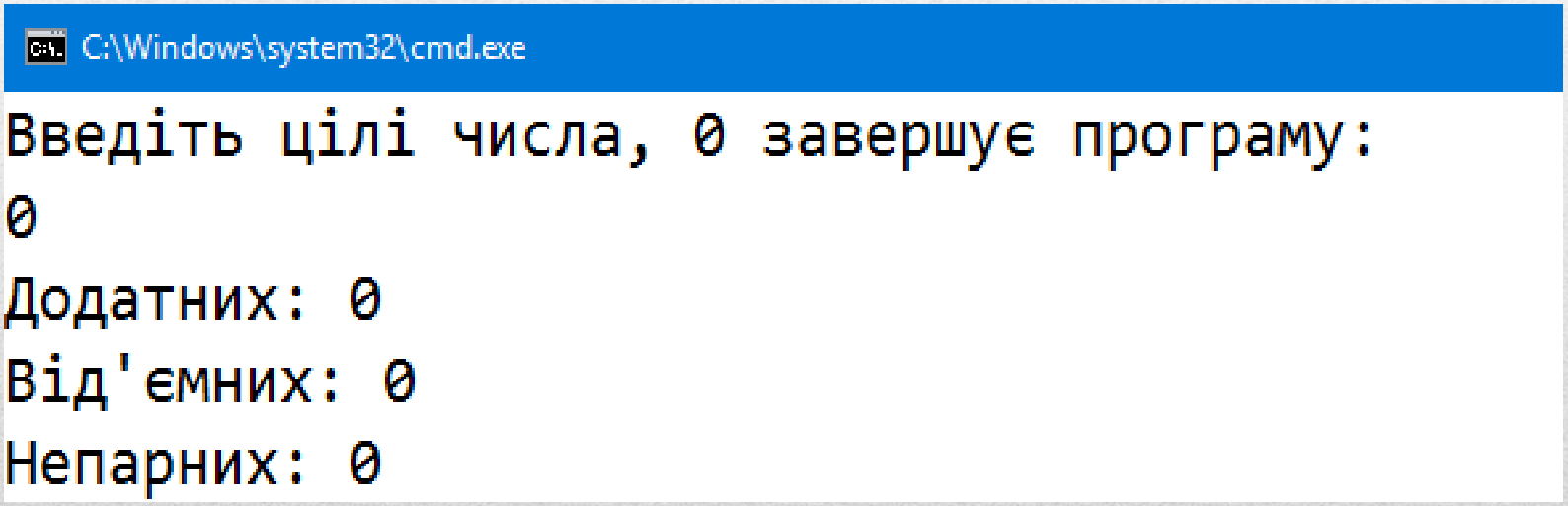
```
int main()
{
    SetConsoleCP(1251);    SetConsoleOutputCP(1251);
    int posCount = 0, negCount = 0, oddCount = 0,
        value;
    printf("Введіть цілі числа, 0 завершує програму:\n");
    scanf("%d", &value);
    while (value != 0)
    {
        if (value > 0) posCount++;
        if (value < 0) negCount++;
        if (value % 2 == 1) oddCount++;
        scanf("%d", &value);
    }
    printf("Додатних: %d\nВід'ємних: %d\nНепарних: %d\n",
        posCount, negCount, oddCount);
    return 0;
}
```

доводиться вводити число перед циклом

Але програма працює правильно:



Якщо одразу ввести **0**, то програма теж правильно працюватиме:



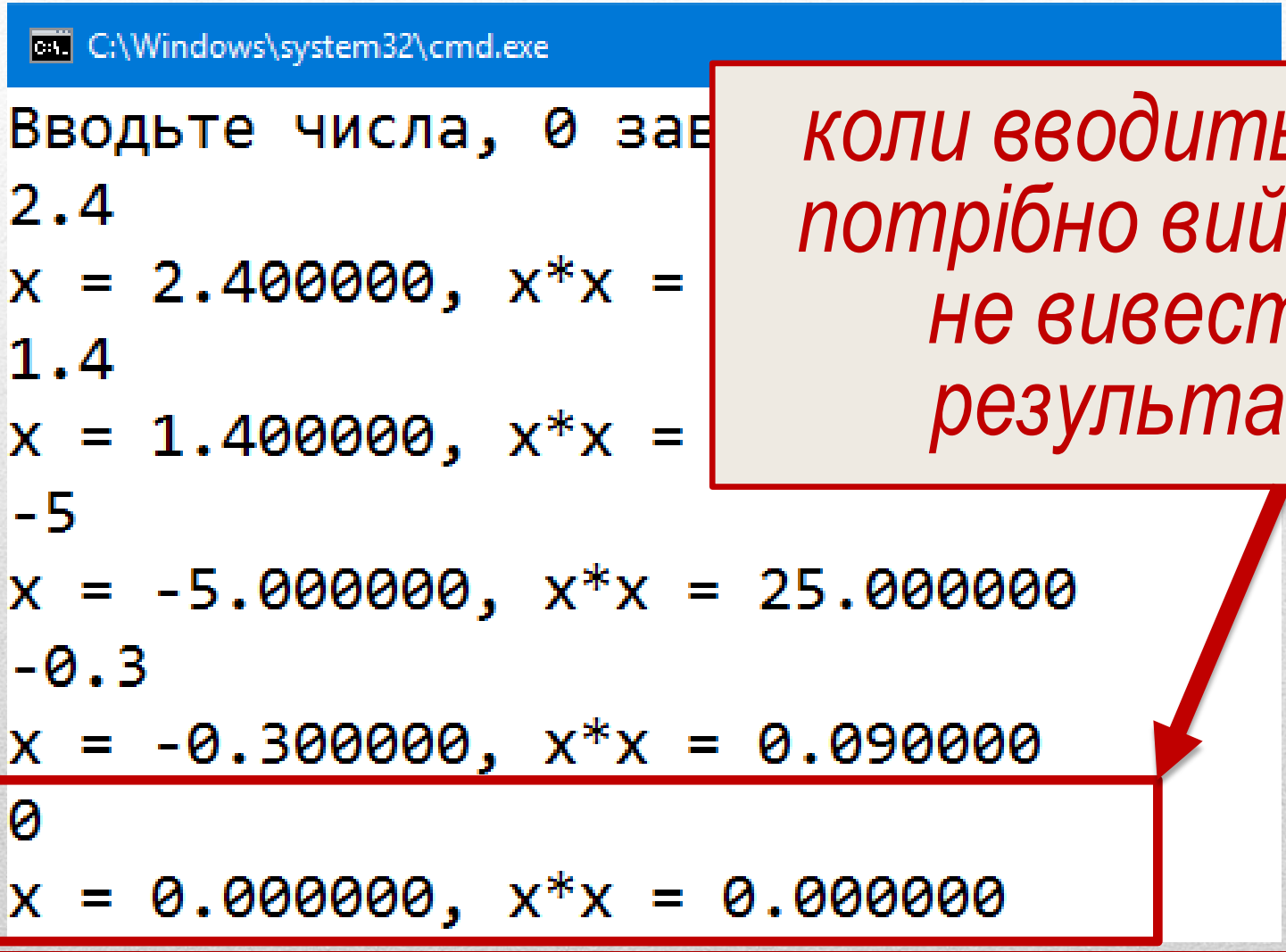
Програма працює так само з циклом do … while:

```
int main()
{
    SetConsoleCP(1251);    SetConsoleOutputCP(1251);
    int posCount = 0, negCount = 0, oddCount = 0,
        value;
    printf("Введіть цілі числа, 0 завершує програму:\n");
    do {
        scanf("%d", &value);
        if (value > 0) posCount++;
        if (value < 0) negCount++;
        if (value % 2 == 1) oddCount++;
    } while (value != 0);
    printf("Додатних: %d\nВід'ємних: %d\nНепарних: %d\n",
         posCount, negCount, oddCount);
    return 0; }
```

Ще одна задача. Написати програму, яка запитує у користувача дійсне число **x** і виводить значення x^2 доти, доки користувач не введе нуль.

```
int main()
{
  SetConsoleCP(1251);  SetConsoleOutputCP(1251);
  double value;
  printf("Вводьте числа, 0 завершує програму:\n");
  do
  {
     scanf("%lf", &value);
     printf("x = %f, x*x = %f\n", value,
                                  value * value);
  } while (value != 0);
  return 0; }
```

Ось результат роботи програми:



коли вводиться 0, потрібно вийти, а не вивести результат

Модифікуємо програму, додаючи ще одну перевірку умови і оператор **break**:

```
int main()
{
  SetConsoleCP(1251);  SetConsoleOutputCP(1251);
  double value;
  printf("Вводьте числа, 0 завершує програму:\n");
  do
  {
     scanf("%lf", &value);
     if (value == 0)
         break;
     printf("x = %f, x*x = %f\n", value,
                                  value * value);
  } while (1);
  return 0;
}
```

?

1	Дано дійсне число - ціна 1 кг цукерок. Вивести вартість 0.1, 0.2, ..., 1 кг цукерок.

```
float c, i = 0;
printf("c:"); scanf("%f", &c);
do {
    i += 0.1;
    printf("c=%f\n", c*i);
} while (i <= 1);
```

```
c:124.57
c=12.46
c=24.91
c=37.37
c=49.83
c=62.28
c=74.74
c=87.20
c=99.66
c=112.11
c=124.57
```

```
float c, i = 0;
printf("c:"); scanf("%f", &c);

while (i <= 1) {
    i += 0.1;
    printf("c=%f\n", c*i);
}
```

?

2	Дано ціле число N (> 0). Знайти суму 1 + 1/2 + 1/3 + ... + 1 / N (дійсне число).

```
int N = 10, j = 1;
float Sum = 0;
do {
    Sum += 1.0 / j;
    j++;
} while (j <=N);
printf("Sum=%f\n", Sum);
```

```
int N = 10, j = 1;
float Sum = 0;
while (j <= N) {
    Sum += 1.0 / j;
    j++;
}
printf("Sum=%f\n", Sum);
```

```
int N = 10;
float Sum = 0;
for (int i = 1; i <=N; i++)
        Sum += 1.0 / j;
printf("Sum=%f\n", Sum);
```

?

3	Дано ціле число N (> 0). Знайти суму N^2 + (N + 1)^ 2 + (N + 2)^ 2 + ... + (N+N)^ 2 (ціле число).

```
int n, rez = 0;
printf("N:");
scanf("%i", &n);

for (int i = 0; i <= n; ++i)
        rez += pow(n + i, 2);
printf("rez = %i \n ", rez);
```

```
int n, rez = 0;
printf("N:");
scanf("%i", &n);
do {
    rez += pow(n + i, 2);
    i ++;
} while (i <= n);
printf("rez = %i \n ", rez);
```

?

4	Дано дійсне число A і ціле число N (> 0). Використовуючи один цикл, знайти суму 1 + A + A^2 + A^3 + ... + A^N.

```
float a, a2 = 1, rez = 1;
int n;
printf("A:"); scanf("%f", &a);
printf("N:");scanf("%i", &n);

for (int i = 0; i <= n; i++)
        rez += pow(a, i);
printf("%f \n ", rez);
```

?

5	Перевірте число x, що ввів користувач, 0< x <10

```
int num;
do {
    printf("введіть число від 0 до 10: ");
    scanf("%d", &num);
} while ((num < 0) || (num > 10));
printf("Введено число= %d\n", num);
```

?

6	Дано ціле число K і набір ненульових цілих чисел; ознака завершення введення чисел - 0. Вивести номер першого числа в наборі, більшого K. Якщо таких чисел немає, то вивести 0.

```
int k, r = 1, num = 0, num1=0, fl=1;
printf("K:");
scanf("%i", &k);
while (r != 0) {
    printf(":"); scanf("%i", &r);

    num++;

    if ((r > k) && (fl == 1)) {
        num1 = num;
        fl = 0;
    }
} printf("%i\n", num1);
```

А можна взагалі скористатися циклом while і операцією **«кома».**

Операція **«кома»** зліва направо прораховує вирази і повертає значення правого операнда.

```
int x = (1, 24);              // x = 24

int y = (1, -24, -3*5); // y = -15
```

Використаємо в операторі **while** операцію **«кома»**

```
int main()
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251);
    double value;
    printf("Вводьте числа, 0 завершує програму:\n");
    while ( scanf("%lf", &value), value != 0 )
        printf("x = %f, x*x = %f\n", value,
            value * value);
}
```

1) спочатку виконується

scanf("%lf", &value)

2) обраховується значення виразу

value != 0

яке і повертається операцією **«,»**

```c
int i = 10, b = 20, c = 30;
i = (b, c);
printf("%i\n", i);

i = (b++, c+b);
printf("%i\n", i);
```

?

```c
int x = 10, y = 20, z = 30;


y=(x, y++, z+y-x);
printf("%d \n ", y);

y=((x--, y--), y-z);

printf("%d \n ", y);
```