Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра «Комп’ютеризовані системи автоматики»



ЗВІТ

про виконання лабораторної роботи №3

«ОПЕРАТОРИ ЦИКЛУ»

із навчальної дисципліни: «Алгоритми і структури даних»

Варіант №1

Виконав:

студент групи ІР-24

Кіндрат Віктор Романович

Прийняв:

к. ф.-м. н., доцент.

Сиротюк С. В.

Львів 2024

**Мета:** ознайомитися з організацією циклічного обчислювального процесу із застосуванням операторів while(), do…while(), for(), навчитися здійснювати контроль вхідних даних при наявності обмежень на їхні значення.

**Програма 3\_1.**

Програма обчислює суму ряду цілих парних чисел, доки сума не стане більше 20, використовуючи оператор while.

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

int main() {

int a=0, sum=0, n=0;

while(sum<20) {

n+=1; // Кількість циклів

a+=2; // Парне число

sum+=a; // Сума парних чисел

printf("n=%d a=%d sum=%d\n",n,a,sum);

}

printf("Loops n=%d\n", n);

system("pause");

return 0;

}

**Завдання до програми 3\_1.**

Розробіть програму, що виконує задані обчислення та виводить на екран результат обчислення, останній член та кількість циклів.

Обчисліть суму ряду цілих чисел, кратних 3, доки останнє число не стане більше 21.

**Програмний код до завдання:**

#include <stdio.h>

int main() {

int sum = 0;

int num = 0;

int cycles = 0;

while (num <= 21) {

num += 3;

sum += num;

cycles++;

}

printf("Summa: %d\n", sum);

printf("Last number: %d\n", num);

printf("Cycles: %d\n", cycles);

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab3 % ./1.out

<- Summa: 108

<- Last number: 24

<- Cycles: 8

**Програма 3\_2.**

Програма обчислює квадрати послідовності чисел.

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

int main() {

int i, b, a=0, kol;

puts("Input number of cycles ");

scanf("%d",&kol); // Введення кількості циклів

for(i=0; i<kol; i++) {

if(a>500) {

puts("a>500"); // Завершення циклу при a>500

break;

}

a+=1; b=a\*a;

printf("a=%3d b=%d\n",a,b);

}

system("pause");

return 0;

}

**Завдання до програми 3\_2.**

Розробіть програму, що виконує задані обчислення та виводить на екран результати (оператор for).

Обчисліть суму та кількість тільки цілих додатних непарних чисел. При введенні числа –11 цикл завершується.

**Програмний код до завдання:**

#include <stdio.h>

int main() {

int num, sum = 0, count = 0, totalNumbers, i;

printf("Введіть загальну кількість чисел: ");

scanf("%d", &totalNumbers);

for (i = 0; i < totalNumbers; i++) {

printf("Введіть число: ");

scanf("%d", &num);

if (num == -11) {

printf("Введено число -11, завершення циклу.\n");

break;

}

if (num > 0 && num % 2 != 0) {

sum += num;

count++;

}

}

printf("Сума додатних непарних чисел: %d\n", sum);

printf("Кількість додатних непарних чисел: %d\n", count);

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab3 % ./2.out

-> Введіть загальну кількість чисел: 5

-> Введіть число: 2

-> Введіть число: 4

-> Введіть число: 3

-> Введіть число: -11

<- Введено число -11, завершення циклу.

<- Сума додатних непарних чисел: 3

<- Кількість додатних непарних чисел: 1

**Програма 3\_3.**

Програма обчислює та виводить на екран таблицю множення, використовуючи вкладений оператор циклу.

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <iostream>

int main() {

int i,j;

for(i=1; i<=10; i++) {

for(j=1; j<=10; j++)

printf("%d \* %d = %2d\n",i,j,i\*j);

printf("\n"); getch();

}

system("pause");

return 0;

}

**Завдання до програми 3\_3.**

Обчисліть та виведіть на екран значення а, b та функції F, використовуючи вкладений оператор циклу, якщо а, b змінюються у заданих діапазонах з заданим кроком.

F=а²+b²; діапазон зміни а від 4 до 20 з кроком 2; діапазон зміни b від 3 до 21 з кроком 3.

**Програмний код до завдання:**

#include <stdio.h>

int main() {

for (int a = 4; a <= 20; a += 2) {

for (int b = 3; b <= 21; b += 3) {

int operation\_res = a \* a + b \* b;

printf("%d^2 + %d^2 = %d\n", a, b, operation\_res);

}

}

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab3 % ./3.out

<- 4^2 + 3^2 = 25

<- 4^2 + 6^2 = 52

<- 4^2 + 9^2 = 97

<- 4^2 + 12^2 = 160

<- 4^2 + 15^2 = 241

<- 4^2 + 18^2 = 340

<- 4^2 + 21^2 = 457

<- 6^2 + 3^2 = 45

<- 6^2 + 6^2 = 72

<- 6^2 + 9^2 = 117

<- 6^2 + 12^2 = 180

<- 6^2 + 15^2 = 261

<- 6^2 + 18^2 = 360

<- 6^2 + 21^2 = 477

<- 8^2 + 3^2 = 73

<- 8^2 + 6^2 = 100

<- 8^2 + 9^2 = 145

<- 8^2 + 12^2 = 208

<- 8^2 + 15^2 = 289

<- 8^2 + 18^2 = 388

<- 8^2 + 21^2 = 505

<- 10^2 + 3^2 = 109

<- 10^2 + 6^2 = 136

<- 10^2 + 9^2 = 181

<- 10^2 + 12^2 = 244

<- 10^2 + 15^2 = 325

<- 10^2 + 18^2 = 424

<- 10^2 + 21^2 = 541

<- 12^2 + 3^2 = 153

<- 12^2 + 6^2 = 180

<- 12^2 + 9^2 = 225

<- 12^2 + 12^2 = 288

<- 12^2 + 15^2 = 369

<- 12^2 + 18^2 = 468

<- 12^2 + 21^2 = 585

<- 14^2 + 3^2 = 205

<- 14^2 + 6^2 = 232

<- 14^2 + 9^2 = 277

<- 14^2 + 12^2 = 340

<- 14^2 + 15^2 = 421

<- 14^2 + 18^2 = 520

<- 14^2 + 21^2 = 637

<- 16^2 + 3^2 = 265

<- 16^2 + 6^2 = 292

<- 16^2 + 9^2 = 337

<- 16^2 + 12^2 = 400

<- 16^2 + 15^2 = 481

<- 16^2 + 18^2 = 580

<- 16^2 + 21^2 = 697

<- 18^2 + 3^2 = 333

<- 18^2 + 6^2 = 360

<- 18^2 + 9^2 = 405

<- 18^2 + 12^2 = 468

<- 18^2 + 15^2 = 549

<- 18^2 + 18^2 = 648

<- 18^2 + 21^2 = 765

<- 20^2 + 3^2 = 409

<- 20^2 + 6^2 = 436

<- 20^2 + 9^2 = 481

<- 20^2 + 12^2 = 544

<- 20^2 + 15^2 = 625

<- 20^2 + 18^2 = 724

<- 20^2 + 21^2 = 841

**Програма 3\_4.**

Програма обчислює суму та добуток десяти непарних значень введених з клавіатури чисел. При введенні числа 0 цикл завершується (break).

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

int main() {

int n=0, n1=0, n2=0, a, sum=0;

long pr=1;

puts("Input value: ");

while(n1<10) {

scanf("%d",&a); // введення значень змінної а

if(a==0) break; // закінчення циклу при а=0

n++; // підрахунок кількості введених значень а

if(!(a%2)) {

n2++; // підрахунок кількості парних значень а

continue;

} // закінчення ітерації

n1++; // підрахунок кількості непарних значень а

sum+=a; pr\*=a; // підрахунок суми та добутку непарних значень а

printf("Number of odd values = %d\n",n1);

printf("sum of odd values %d\n",sum);

printf("Work odd values %ld\n",pr);

}

printf("Number of input values = %d\n",n);

printf("Number of even values = %d \n",n2);

system("pause");

return 0;

}

**Завдання до програми 3\_4.**

Розробіть програму, що виконує задані обчислення з числами, які вводяться з клавіатури, та виводить на екран результати обчислень.

Знайдіть найбільше число з введених 10 цілих чисел та обчисліть суму, кількість чисел, що більші 5. При введенні числа 10 цикл завершується (оператор break).

**Програмний код до завдання:**

#include <stdio.h>

int main() {

int sum\_of\_more\_than\_5 = 0, count\_of\_more\_than\_5 = 0;

int max;

for (int i = 1; i <= 10; i++) {

int current;

printf("Input a number %d:\n", i);

scanf("%d", ¤t);

if (current == 10) {

if (i == 1) {

max = 10;

}

break;

}

if (current > 5) {

sum\_of\_more\_than\_5 += current;

count\_of\_more\_than\_5++;

}

if (current > max) {

max = current;

}

}

printf("Max number is = %d\n", max);

printf("Sum of more than 5 = %d\n", sum\_of\_more\_than\_5);

printf("Count of more is = %d\n", count\_of\_more\_than\_5);

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab3 % ./4.out

<- Input a number 1:

-> 5

<- Input a number 2:

-> 6

<- Input a number 3:

-> 7

<- Input a number 4:

-> 9

<- Input a number 5:

-> 2

<- Input a number 6:

-> 10

<- Max number is = 9

<- Sum of more than 5 = 22

<- Count of more is = 3

**Програма 3\_5.**

В програмі використовується оператор do…while() для контролю вхідних даних, при введенні числа х в заданому діапазоні обчислюються функції acos(х), asin(х).

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <math.h>

int main() {

double x;

int kol=0;

do {

puts("Input value -1<=x<=1");

scanf("%lf",&x);

kol++; // підрахунок кількості введених значень

} while(x<-1.0||x>1.0);

printf("acos(%.2f) = %8.5f\n",x,acos(x));

printf("asin(%.2f) = %8.5f\n",x,asin(x));

printf("Input %d value\n",kol);

system("pause");

return 0;

}

**Завдання до програми 3\_5.**

Розробіть програму, що виконує обчислення з числами, які вводяться з клавіатури (оператор for) і знаходяться в заданому діапазоні (оператор do…while), та виводить на екран результати обчислень і кількість введених та обчислених чисел.

Обчисліть синус та косинус (sin(х), cos(х)) тих п’яти дійсних чисел, що знаходяться в діапазоні від 0 до 2π.

**Програмний код до завдання:**

#include <stdio.h>

#include <math.h>

int main() {

int totalInputs = 0;

const int maxInputs = 12;

for (int i = 1; i <= 5; i++) {

double x;

int kol = 0;

do {

puts("Input value 0 <= x <= 2PI:");

scanf("%lf", &x);

if (x >= 0 && x <= 2 \* M\_PI) {

printf("sin(%.2f) = %8.5f\n", x, sin(x));

printf("cos(%.2f) = %8.5f\n", x, cos(x));

kol++;

totalInputs++;

} else {

puts("Input is out of range! Please enter a value between 0 and 2PI.");

}

if (totalInputs >= maxInputs) {

return -1;

}

} while (kol < 1);

}

printf("Total valid inputs entered: %d\n", totalInputs);

return 0;

}

**Приклад роботи програмного коду:**

viktorkindrat@MacBook-Air-Viktor lab3 % ./5.out

<- Input value 0 <= x <= 2PI:

-> 10

<- Input is out of range! Please enter a value between 0 and 2PI.

<- Input value 0 <= x <= 2PI:

-> 1

-> sin(1.00) = 0.84147

-> cos(1.00) = 0.54030

<- Input value 0 <= x <= 2PI:

-> 2

-> sin(2.00) = 0.90930

-> cos(2.00) = -0.41615

<- Input value 0 <= x <= 2PI:

-> 3

-> sin(3.00) = 0.14112

-> cos(3.00) = -0.98999

<- Input value 0 <= x <= 2PI:

-> 4

-> sin(4.00) = -0.75680

-> cos(4.00) = -0.65364

<- Input value 0 <= x <= 2PI:

-> 5

-> sin(5.00) = -0.95892

-> cos(5.00) = 0.28366

-> Total valid inputs entered: 5

**Висновок:**

У результаті виконання лабораторної роботи я розширив свої знання із мови програмування C. Ознайомився з організацією циклічного обчислювального процесу із застосуванням операторів while(), do…while(), for(), навчився здійснювати контроль вхідних даних при наявності обмежень на їхні значення. У результаті виконання написаного програмного коду отримав очікувані значення.