

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 1

з дисципліни “ Основи програмування ”

## тема “Обчислення математичних формул. Використання конструкцій розгалуження та циклів ”

|  |  |  |
| --- | --- | --- |
| Виконала  студентка I курсу  групи КП-61  Чумак Надія Русланівна  (*прізвище, ім’я, по батькові*)  варіант №\_\_\_\_\_\_\_ |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Штрафні бали:   |  |  | | --- | --- | | **Термін здачі** | **Оформлення звіту** | |  |  | | Нараховані бали:   |  |  |  | | --- | --- | --- | | **Корект. програм (2 бала)** | **Відп. на теор. питання (1 бал)** | **Відп. на прогр. питання (2 бала)** | |  |  |  | | Сумарний бал:   |  | | --- | |  | |

Київ 2016

**Мета роботи**

Навчитися на практиці проводити точні обчислення математичних формул за допомогою операторів та функцій мови програмування С.

Застосувати теоретичні знання для створення програмного забезпечення для прийняття рішень на основі вхідних даних за допомогою умовних конструкцій та конструкцій розгалуження.

Застосувати на практиці різні види циклічних конструкцій для обчислення математичних формул.Навчитися виконувати компіляцію та базове тестування власного коду за допомогою компілятора мови С та спеціалізованої IDE.

**Постановка завдання**

**Завдання 1.** Виконати розрахунки за заданими формулами.

*,*де

  - довільні дробові числа, які користувач вводить із консолі на початку роботи програми.Необхідна точність обчислень  .У випадку неіснування відповіді виводити відповідне повідомлення у консоль.

**Завдання 2.** Користувач вводить 3 цілочисельні значення a, b i c.

На основі вхідних даних і заданих умов за допомогою умовних конструкцій (if…else) визначити істинність результату і вивести його у консоль.

|  |
| --- |
| Якщо всі введені числа a, b та c від’ємні, то: Знайти значення modmin: модуль найменшого числа; Знайти значення sum2: суму двох інших чисел; Якщо sum2 більша -256, а modmin є степенню двійки, то: Результат: True Інакше: Результат: True якщо модуль sum2 більше 16 або modmin більше 8;  Якщо серед введених чисел є і від’ємні і додатні (або 0), то: Якщо тільки одне з них від’ємне, то: Результат: чи це від’ємне число більше -256; Якщо два від’ємних числа, то Результат: чи сума цих двох від’ємних чисел помножена на 11 більша за -256; Якщо всі числа додатні (або 0) то: Знайти значення max: максимальне з цих чисел; Знайти значення min: мінімальне з цих чисел; Результат: чи max в степені min можна зберегти у змінну **short** **int**; |

У деяких пунктах результатом буде відповідь на питання представлена булевими значеннями.

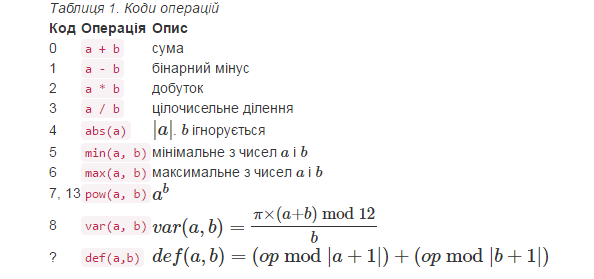
Наприклад:

"Результат: чи це від’ємне число більше -256"?:

* "- Так" - це значення True
* "- Ні" - це значення False

**Завдання 3.** У таблиці 1 задано коди операцій, відповідні операції та їх

детальний опис. Користувач вводить код операції (значення op) та два цілочисленних значення a i b.Якщо значення коду op від’ємне - міняти місцями значення a i b, а op змінити на додатнє. Використовуючи конструкції розгалуження (switch) обчислити та вивести результат відповідної операції. Якщо значення обчислити неможливо - виводити про це повідомлення у консоль.



**Завдання 4.** Користувач вводить 2 додатні цілочисельні значення n i m.

За допомогою ітераційних циклічних конструкцій (for) обчислити значення x та вивести його консоль:

**Тексти коду програм**

**Завдання 1.**

|  |
| --- |
| math.c |
| #include <math.h>  #include <progbase.h>  int main(void)  {  double x = 0;  double y = 0;  double z = 0;  double a1 = 0;  double a2 = 0;  double a3 = 0;  double a = 0;      puts ("Введіть цілочисельне x:");  x = getDouble();  printf ("x = %f\n" , x );    puts ("Введіть цілочисельне y:");  y = getDouble();  printf ("y = %f\n" , y );    puts ("Введіть цілочисельне z:" );  z = getDouble ();  printf ("z = %f\n" , z );    if ( !fabs(x+y)) {  printf ("Неможливо обчислити значення a0 \n");  }  else {  a2 = y / (10 \*fabs(x + y));  printf ( "a0 = %.10f\n" , a2 );  }  if ( !z ) {  printf ("Неможливо обчислити значення a1 \n");  } else if ((x - y) <= 0) {  printf ("Неможливо обчислити значення a1 \n");  }  else {  a1 = pow(x, y+1) / pow(x-y, 1/z);  printf ( "a1 = %.10f\n" , a1 );  }  if ( !(z + fabs(x-y))) {  printf ("Неможливо обчислити значення a2\n");  }  else {  a3 = pow(2 + sin(y),cos(x)/(z + fabs(x-y)));  printf ( "a2 = %.10f\n" , a3 );  }  if ( !(z + fabs(x-y))) {  printf ("Неможливо обчислити значення a \n");  } else if ( !fabs(x+y) ) {  printf ("Неможливо обчислити значення a \n");  }  else if ( (x - y) <= 0) {  printf ("Неможливо обчислити значення a \n");  }  else if ( !z ) {  printf ("Неможливо обчислити значення a \n");  }  else {  a = a1 + a2 + a3;  printf ( "a = %.10f\n" , a );  }    return 0;  } |

**Завдання 2.**

|  |
| --- |
| condition.c |
| #include <math.h>  #include <progbase.h>  int main(void)  {  int a = 0;  int b = 0;  int c = 0;  int modmin = 0;  int sum2 = 0;  int min = 0;  int max = 0;    puts ( "Enter int a:" );  a = getInt();  printf ( "Your: a = %d\n" , a );    puts ( "Enter int b:" );  b = getInt();  printf ( "Your : b = %d\n" , b );    puts ( "Enter int c:" );  c = getInt();  printf ( "Your: c = %d\n" , c );    if (a < 0 && b < 0 && c < 0) {  int min = a ;  if (b < min ) {  min = b;  modmin = abs(b);  sum2 = a + c;    }  if (c < min) {  min = c;  modmin = abs(c);  sum2 = a + b;  }  {  modmin = abs (a);  sum2 = b + c;  }  while (modmin % 2 == 0)  modmin = modmin/2;  if (modmin == 1,sum2 > -256) {  printf ( "true \n" );  }  else {  if ( abs(sum2) > 16 ) {  printf ( "true \n");  } else if ( modmin > 8 ) {  printf ( "true \n");  }  printf ("false \n");    }  }  }  if (a >= 0 || b >= 0 || c >= 0 ) {  if (a < 0 ) {  if ( a > -256) {  printf ( "true \n");  }  else {  printf ( "false \n");  }  }  if (b < 0 ) {  if ( b > -256) {  printf ( "true \n");  }  else {  printf ( "false \n");  }  }  if (c < 0 ) {  if ( c > -256) {  printf ( "true \n");  }  else {  printf ( "false \n");  }  }  if ( (a < 0 && b < 0) || ( a < 0 && c < 0) || (b < 0 && c < 0) ) {  if ((11\*(a + b) > -256) || (11\*(b + c) > -256) || (11\*(a + c) > -256)) {  printf ( "true \n");  }  else {  printf ( "false \n");  }  }  }        if (a >= 0 && b >= 0 && c >= 0) {  int min = a ;  int max = a;  if (b <= min ) {  min = b;  }  if (c <= min) {  min = c;  }  if (b >= max) {  max = b;  }  if (c >= max) {  max = c;  }  if (pow(max,min) >= -32768 && pow(max,min) <= 32767){  printf ( "true \n" );  }  else {  printf ( "false \n" );  }    }  printf("\n");    return 0;  } |

**Завдання 3.**

|  |
| --- |
| switch.c |
| #include <math.h>  #include <progbase.h>  int main(void)  {  int a = 0;  int b = 0;  int op = 0;  int temp = 0;  int min = 0;  int max = 0;  int sum = 0;  float product = 0;  float div = 0;  float c = 0;  const double PI = 3.14159265;  puts ("Введіть цілочисельне a:" );  a = getInt();  printf ("a = %d\n" , a );  puts ("Введіть цілочисельне b:" );  b = getInt();  printf ("b = %d\n" , b );    printf ("0 сума \n");  printf ("1 бінарний мінус : \n");  printf ("2 добуток \n");  printf ("3 цілочисельне ділення \n");  printf ("4 |a| \n");  printf ("5 мінімальне з чисел a i b \n");  printf ("6 максимальне з чисел a i b \n");  printf ("7 a^b\n");  printf ("8 var(a,b) = (pi\*(a + b) mod 12)/2 \n");  printf ("? def(a,b) = (op mod|a + 1| op mod |b + 1|) \n");      puts ("Введіть код операції(цілочисельне значення op):" );  op = getInt();  printf ("op = %d\n" , op );    if ( op < 0 ) {  temp = a;  a = b;  b = temp;  op = abs(op);  }  switch(op) {  case 0:{  c = a + b;  printf ("сума : %f\n" , c);  }  case 1: {  c = a - b;  printf ("бінарний мінус : %f\n" , c );  }  break;  case 2: {  c = a\*b;  printf ("добуток : %f\n" , c );  }  break;  case 3: {  if ( !b ) {  printf ("Неможливо обчислити значення\n");  }  else {  c = a/b;  printf ("цілочисельне ділення : %f\n" , c );  }  }  break;  case 4: {  c = abs(a);  printf ( "%f\n" , c );  }  break;  case 5:{  if ( a < b ) {  min = a;  printf ("мінімальне з чисел a i b : %d\n" , min );  }  else {  min = b;  printf ("мінімальне з чисел a i b : %d\n" , min );  }  }  break;  case 6:{  if ( a > b ) {  max = a;  printf ("максимальне з чисел a i b : %d\n" , max );  }  else {  max = b;  printf ("максимальне з чисел a i b : %d\n" , max );  }  }  break;  case 7: {  if ( a > 0 ) {  c = pow(a,b);  printf ("a %f\n" , c);  }  else {  printf ("Неможливо обчислити значення\n");  }  }  break;  case 8: {  if ( !b ) {  printf ("Неможливо обчислити значення\n");  }  else {  c = (PI\*((a+b) % 12))/b;  printf ("var(a,b) = %f\n" , c);  }  }  break;  default: {  c = ( op%abs(a + 1) + op%abs(b + 1));  printf ("def(a,b) = %f\n" , c);  }  }  return 0;  } |

**Завдання 4.**

|  |
| --- |
| loop.c |
| #include <math.h>  #include <progbase.h>  int main(void)  {  int n = 0;  int m = 0;  int i = 0;  int j = 0;  double sum1 = 0;  double sum2 = 0;  const double PI = 3.14159265;    puts ( "Enter int m:" );  m = getInt();  printf ( "Your: m = %d\n" , m );  puts ( "Enter int n:" );  n = getInt();  printf ( "Your : n = %d\n" , n );      for (i = 1; i <= n; i++) {  sum2 = 0;  for (j = 1; j <= m; j++) {  sum2 += (cos(i\*PI/2) + 3 \* sin (j\*PI/2));  }  sum1 += sum2;  }  printf ( "x = %f\n" , sum1 );    return 0;  } |

**Приклади результатів**

**Завдання 1.**

Виконання розрахунків за формулою у правильному форматі

|  |
| --- |
| Введіть цілочисельне x:  6  x = 6.000000  Введіть цілочисельне y:  7  y = 7.000000  Введіть цілочисельне z:  4  z = 4.000000  a0 = 0.0538461538  Неможливо обчислити значення a1  a2 = 1.2064163419  Неможливо обчислити значення a |

Виконання розрахунків за формулою у неправильному форматі

|  |
| --- |
| Введіть цілочисельне x:  6  x = 6.000000  Введіть цілочисельне y:  7  y = 7.000000  Введіть цілочисельне z:  4  z = 4.000000  a0 = 0.0538461538  Неможливо обчислити значення a1  a2 = 1.2064163419  Неможливо обчислити значення a |

**Завдання 2.**

Визначення істинності результату за допомогою оператора (if…else)

|  |
| --- |
| Enter int a:  -4  Your: a = -4  Enter int b:  -5  Your : b = -5  Enter int c:  -6  Your: c = -6  true |

Визначення істинності результату за допомогою оператора (if…else)

|  |
| --- |
| Enter int a:  34  Your: a = 34  Enter int b:  23  Your : b = 23  Enter int c:  56  Your: c = 56  false |

**Завдання 3.**

Використання конструкції розгалуження (switch) для обчислення відповідної операції.

|  |
| --- |
| Введіть цілочисельне a:  3  a = 3  Введіть цілочисельне b:  4  b = 4  0 сума  1 бінарний мінус :  2 добуток  3 цілочисельне ділення  4 |a|  5 мінімальне з чисел a i b  6 максимальне з чисел a i b  7 a^b  8 var(a,b) = (pi\*(a + b) mod 12)/2  ? def(a,b) = (op mod|a + 1| op mod |b + 1|)  Введіть код операції(цілочисельне значення op):  7  op = 7  a^b = 81.000000 |

Використання конструкції розгалуження (switch) для обчислення відповідної операції у випадку введення від’ємного op

|  |
| --- |
| Введіть цілочисельне a:  4  a = 4  Введіть цілочисельне b:  5  b = 5  0 сума  1 бінарний мінус :  2 добуток  3 цілочисельне ділення  4 |a|  5 мінімальне з чисел a i b  6 максимальне з чисел a i b  7 a^b  8 var(a,b) = (pi\*(a + b) mod 12)/2  ? def(a,b) = (op mod|a + 1| op mod |b + 1|)  Введіть код операції(цілочисельне значення op):  -6  максимальне з чисел a i b : 5 |

**Завдання 4.**

Виконання ітераційних циклічних конструкцій (for) для обчислення значення x

|  |
| --- |
| Enter int m:  3  Your: m = 3  Enter int n:  2  Your : n = 2  x = -3.000000 |

**Висновки**

Виконавши лабораторну роботу, навчилися проводити точні обчислення математичних формул за допомогою операторів та функцій мови програмування С, використовуючи теоретичні та практичні знання.Мова С надає засоби достатньо точної обробки числових типів даних для широкого застосування.

Здатність управляти програмним потоком дозволяє робити вибіркове виконання окремих ділянок коду, а це дуже цінна особливість програмування для демонстрації спрощеного програмного коду, що використовується у завданнях 1 та у завданні 2.

Використовуючи конструкції розгалуження, що були продемонстровані у завданні 3, залежно від умови виконували різні дії, результат яких записаний у таблиці.

За допомогою циклічного оператора for, що використовується для створення циклів, що виконуються задану кількість разів, досягли бажаного результату у завданні 4 і вивели кінцеве значення у консоль

Компіляція всього коду відбувалася за допомогою компілятора gcc. Робота програм була протестована за допомогою допоміжного програмного коду. Код є валідним і компілюється компілятором С із певними заданими опціями без помилок .