

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”

|  |  |  |
| --- | --- | --- |
| Виконав  студент I курсу  групи КП-62  Лук’янець Михайло Олександрович  (*прізвище, ім’я, по батькові*)  варіант №15 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 2016 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. Реалізація математичних операцій**

Виконати розрахунки за заданими формулами.

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

**Завдання №2. Умовні конструкції**

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.

**Завдання №3. Конструкції розгалуження**

У таблиці 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. Використання циклічних конструкцій**

Користувач вводить 2 додатні цілочисельні значення n i m.

За допомогою ітераційних циклічних конструкцій (for) обчислити значення x та вивести його консоль за заданою формулою.

**Тексти коду програм**

**Завдання 1.**

|  |
| --- |
| main.c |
| #include <stdio.h>  #include <stdlib.h>  #include <progbase.h>  #include <math.h>  int main()  {  double x, y, z, a0, a1, a2;  printf("Please, enter a variable(double) x:\n");  x = getDouble();  printf("Please, enter a variable(double) y:\n");  y = getDouble();  printf("Please, enter a variable(double) z:\n");  z = getDouble();  if((x - y == 0) || (z == 0) || (x + y == 0) || (sin(x) == 0) || ( x - y < 0))  {  printf("Wrong numbers. It won't work. Trust me. I'm an engineer");  return 1;  }  a0 = pow( x, y + 1) / pow( x - y, 1/z ) ;  a1 = y / ( 6 \* fabs(x + y));  a2 = sqrt( fabs( ( cos(y) / sin(x)) + 1 ));  printf("Result: %.10f", a0 + a1 + a2);  return 0;  } |

**Завдання 2.**

|  |
| --- |
| main.c |
| #include <stdio.h>  #include <stdlib.h>  #include <progbase.h>  #include <math.h>  int testNumber( int fubar);  int main()  {  int a, b , c, modmin, sum2, min, max;  int flag = 0;  int counter = 0;  int res1 = 0;  printf("Please, enter a variable(int) a:\n");  a = getInt();  printf("Please, enter a variable(int) b:\n");  b = getInt();  printf("Please, enter a variable(int) :\n");  c = getInt();  if(a < 0 && b < 0 && c < 0)  {  if(a < b)  {  if(a < c)  {  modmin = abs(a);  sum2 = b + c;  }  else  {  modmin = abs(c);  sum2 = a + b;  }  }  else if( b < c)  {  modmin = abs(b);  sum2 = a + c;  }  else  {  modmin = abs(c);  sum2 = a + b;  }  if(modmin != 0 && !(modmin&(modmin-1))){  flag = 1;  }  if( sum2 > -256 && flag == 1)  {  printf("Result: True\n");  }  else if( (abs(sum2) - modmin < 16) || sum2 > -16 )  {  printf("Result: True\n");  }  else  {  printf("Result: False\n");  }  }  else if( a < 0 || b < 0 || c < 0 )  {  if( a < 0)  {  counter ++;  res1 += a;  }  if( b < 0)  {  counter ++;  res1 += b ;  }  if( c < 0)  {  counter ++;  res1 += c ;  }  if(counter == 2 && ( res1\*6 > -256 ))  {  printf("Result: True\n");  }  else if( res1 > -256)  {  printf("Result: True\n");  }  else  {  printf("Result: False\n");  }  }  else  {  if(a < b)  {  if(a < c)  {  min = abs(a);  }  else  {  min = abs(c);  }  }  else if( b < c)  {  min = abs(b);  }  else  {  min = abs(c);  }  if(a > b)  {  if(a > c)  {  max = a;  }  else  {  max = c;  }  }  else if( b > c)  {  max = b;  }  else  {  max = c;  }  if( max - min > 32)  {  printf("Result: True\n");  }  else  {  printf("Result: False\n");  }  }  return 0;  } |

**Завдання 3.**

|  |
| --- |
| main.c |
| #include <stdio.h>  #include <stdlib.h>  #include <progbase.h>  #include <math.h>  int main()  {  int a, b , op;  int result = 0;  printf("Please, enter a variable(int) a:\n");  a = getInt();  printf("Please, enter a variable(int) b:\n");  b = getInt();  printf("Please, enter what you want to do: \n0 a + b\n1 a - b \n2 a \* b\n3 a / b\n4 abs(a)\n5 min(a, b)\n");  printf("6 max(a, b)\n7, 13 pow(a, b)\n8 var(a, b)\nelse def(a,b)\n op < 0 to reverse a and b\n");  op = getInt();  if(op < 0)  {  a = a + b;  b = a - b;  a = a - b;  op = abs(op);  }  switch( op)  {  case 0:  {  result = a + b;  }  break;  case 1:  {  result = a - b;  }  break;  case 2:  {  result = a \* b;  }  break;  case 3:  {  if(b == 0){  printf("Wrong numbers. It won't work. Trust me. I'm an engineer\n");  break;  }  result = a / b;  }  break;  case 4:  {  result = abs(a);  }  break;  case 5:  {  result = fmin(a, b);  }  break;  case 6:  {  result = fmax(a, b);  }  break;  case 7  case 13 :  {  result = pow (a, b);  }  break;  case 8:  {  if(b == 0){  printf("Wrong numbers. It won't work. Trust me. I'm an engineer\n");  break;  }  result = ( (int)((3.14) \* ( a + b)) % 7) / b;  }  break;  default:  {  result = op \* abs(a + 1) + op \* abs(b + 1);  }  break;  }  printf("Result: %i", result);  return 0;  } |

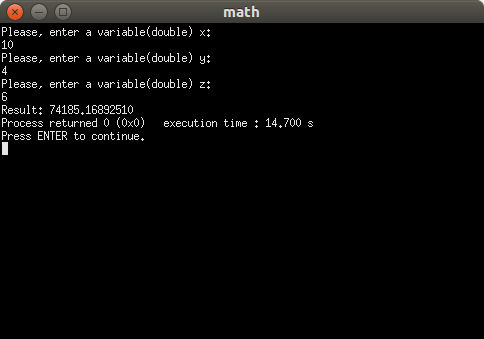
**Завдання 4.**

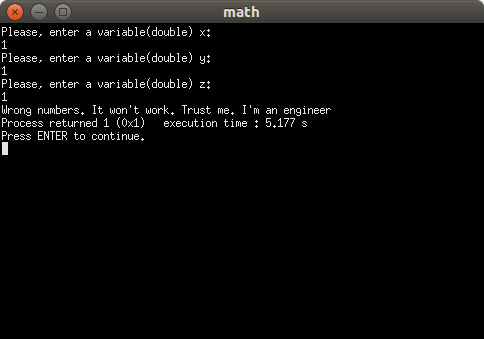
|  |
| --- |
| main.c |
| #include <stdio.h>  #include <stdlib.h>  #include <progbase.h>  #include <math.h>  int main()  {  int m, n , i;  double result = 0.0;  int semiResult = 0.0;  double cosSemiRes = 0.0;  printf("Please, enter a variable(int) m:\n");  m = getInt();  printf("Please, enter a variable(int) n:\n");  n = getInt();  for( i = 0; i < m; i++){  semiResult += pow((i + 1), 2);  }  cosSemiRes = cos(semiResult);  for(i = 0; i < n; i++){  result += (sin((2\*3.14) / (i + 2) )) \* cosSemiRes;  }  printf("Result: %f\n", result);  return 0;  } |

**Приклади результатів**

**Завдання 1.**

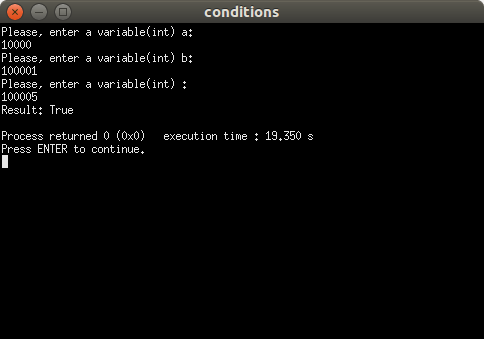
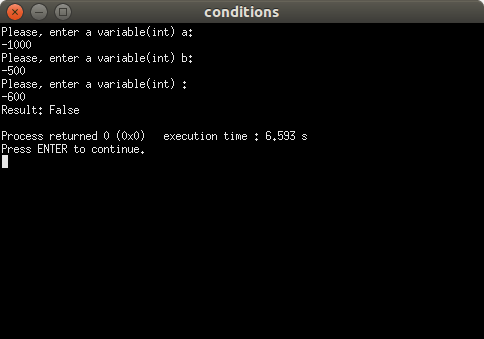
Приклади правильного(1) та неправильного(2) вводу початкових данних.





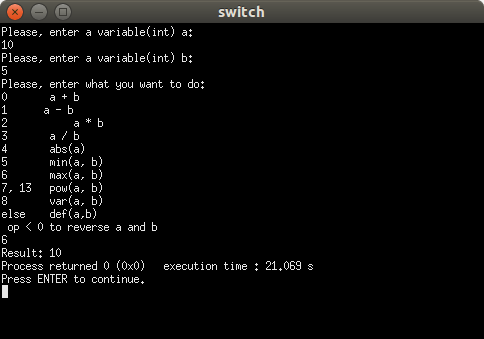
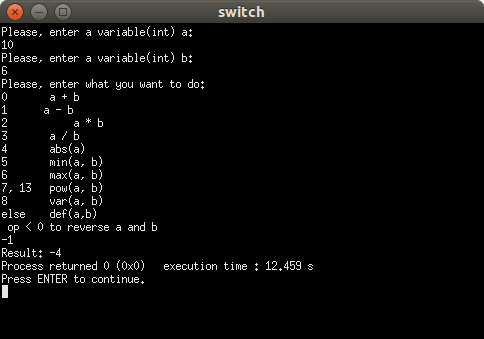
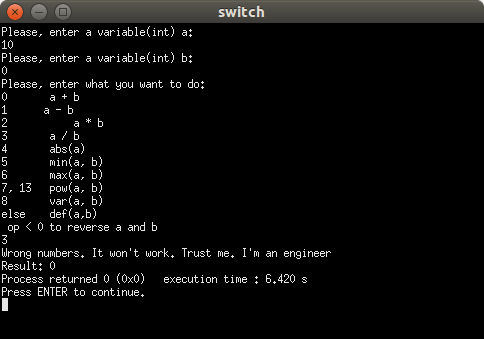
**Завдання 2.**

Результати при введенні різних даних



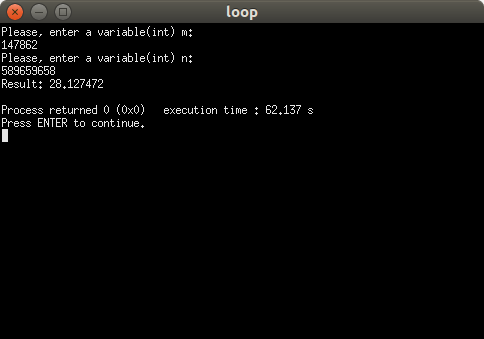
**Завдання 3.**

Результати при введенні правильних (2,3) та неправильних(1) даних.



**Завдання 4.**

Приклад результату при введенні даних.



**Висновки**

Виконавши дану лабораторну роботу було проведено точні обчислення математичних формул за допомогою операторів та функцій мови С. Мова С надає засоби достатньо точної обробки числових типів даних для широкого застосування.

Також, за допомогою конструкцій розгалуження, можливо виконувати різні обрахунки на основі вхідних даних та задавати у програмі умови обробки даних і прийняття рішень на основі них, що й було показано на практиці у завданнях 2 і 3.

В результаті виконання завдання 3 було застосовано конструкції розгалуження для демонстрації спрощення програмного коду при прийнятті програмою рішень і обробки числових значень на основі вхідних даних.

За допомогою циклічних операторів можливо проводити багаторазові обчислення, що було продемонстровано у завданні 4.

Компіляція всього коду відбувалася за допомогою компілятора gcc. Робота програм була протестована за допомогою допоміжного програмного коду, текст якого наведено у додатку А.