

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(ла)  студент(ка) I курсу  групи КП-61  Корунська Анна Михайлівна  (*прізвище, ім’я, по батькові*)  варіант № 17 |  | 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 або спеціалізованої IDE.

**Постановка завдання**

**Завдання 1.** Виконати розрахунки за заданими формулами. У випадку неіснування відповіді повертати константу **NAN** з <math.h>. Необхідна точність обчислень .

=++**,** де

**Завдання 2.** Написати алгоритм, яка на основі вхідних даних і заданих умовних правил визначає істинність результату.

**Завдання 3.** Задано арифметичні операції і відповідні їм коди. На вхід подається код операції та два цілочисленних значення. Використовуючи конструкції розгалуження обчислити та повернути результат операції.

**Завдання 4.** На вхід подаються значення m і n. За допомогою циклічних конструкцій обчислити значення х, заданого формулою.

**Тексти коду програм**

**Завдання 1.**

|  |
| --- |
| main.c |
| **#include <stdio.h>**  **#include <progbase.h>**  **#include <math.h>**  **int main(void)**  **{**  **double a = 0.0;**  **double a0 = 0.0;**  **double a1 = 0.0;**  **double a2 = 0.0;**  **double x = 0.0;**  **double y = 0.0;**  **double z = 0.0;**    **puts("Enter double x:");**  **x = getDouble();**  **printf("You entered: x = %.10f\n", x);**  **puts("Enter double y:");**  **y = getDouble();**  **printf("You entered: y = %.10f\n", y);**  **puts("Enter double z:");**  **z = getDouble();**  **printf("You entered: z = %.10f\n", z);**  **printf("\n Results: \n");**    **if( x - y != 0 && z!= 0 && (x - y) > 0) {**  **a0 = (pow(x, y + 1)) / sqrt(x - y);**  **printf("a0 = %.10f\n", a0);**  **} else {**  **printf("Answer can't be computed, a0 is not a number.\n");**  **}**    **if((x + y)!= 0){**  **a1 = y / (8 \* fabs(x + y));**  **printf("a1 = %.10f\n", a1);**  **} else {**  **printf("Answer can't be computed, a1 is not a number.\n");**  **}**    **if(sin(x) != 0){**  **a2 = sqrt(fabs((cos(y) / sin(x)) + 1));**  **printf("a2 = %.10f\n", a2);**  **} else {**  **printf("Answer can't be computed, a2 is not a number.\n");**  **}**    **if( x - y != 0 && z!= 0 && (x - y) > 0 && (x + y)!= 0 && sin(x) != 0) {**  **a = a0 + a1 + a2;**  **printf("a = %.10f\n", a);**  **} else {**  **printf("a0 can't be computed\n");**  **}**    **return 0;**  **}** |

**Завдання 2.**

|  |
| --- |
| main.c |
| **#include <stdio.h>**  **#include <math.h>**  **#include <progbase.h>**  **int main(void) {**  **int a = 0;**  **int b = 0;**  **int c = 0;**  **int min = 0;**  **int max = 0;**  **int sum = 0;**  **int sum2 = 0;**  **int result = 0;**  **int modmin = 0;**    **puts("Enter int a:");**  **a = getInt();**  **printf("You entered: a = %i\n", a);**  **puts("Enter int b:");**  **b = getInt();**  **printf("You entered: b = %i\n", b);**  **puts("Enter int c:");**  **c = getInt();**  **printf("You entered: c = %i\n", c);**    **sum = a + b + c;**  **min = fmin( a, fmin(b, c));**  **max = fmax( a, fmax(b, c));**    **if( a < 0 && b < 0 && c < 0){**  **modmin = fabs(min);**  **printf("modmin = %i\n", modmin);**  **sum2 = sum - min;**  **printf("sum of two other values = %i\n", sum2);**  **if( sum2 > -256 && fmod(log2(modmin), 1) == 0){**  **result = 1;**  **} else if ( fabs(sum2) - modmin < 16 || fabs(sum2) < 16) {**  **result = 1;**  **}**    **if(result == 0) {**  **printf("Result is false\n");**  **} else {**  **printf("Result is true\n");**  **}**  **}**    **else if(fabs(sum) > sum) {**  **if((a + b + c - min) == fabs(a + b + c - min))/\* если два числа положительные \*/{**  **if(min > - 256) {**  **result = 1;**  **} else {**  **if((sum - max)\*8 > -256) {**  **result = 1;**  **}**  **}**  **}**  **if(result == 0) {**  **printf("Result is false\n");**  **} else {**  **printf("Result is true\n");**  **}**  **}**    **else {**  **if( max\*min < 256){**  **result = 1;**  **}**    **if(result == 0) {**  **printf("Result is false\n");**  **} else {**  **printf("Result is true\n");**  **}**  **}**    **return 0;**  **}** |

**Завдання 3.**

|  |
| --- |
| main.c#include <stdio.h>  #include <math.h>  #include <progbase.h>  int main(void) {  int a = 0;  int b = 0;  int op = 0;  int temp = 0;  double result = 0;  const double pi = 3.1415926535;    printf("List of possible operations:\  \n 0\ta + b \n 1\ta - b \n 2\ta \* b \n 3\ta / b \n 4\tabs(a) \n 5\tmin(a, b) \n 6\tmax(a, b) \n 7, 13\tpow(a, b)\  \n 8\tvar(a, b) \n other\tdef(a, b)\n");    puts("\nEnter operation code:");  op = getInt();  printf("You entered operation code %i\n", op);    puts("\nEnter int a:");  a = getInt();  printf("You entered: a = %i\n", a);  puts("Enter int b:");  b = getInt();  printf("You entered: b = %i\n", b);  if(op < 0){  op = abs(op);    temp = a;  a = b;  b = temp;    printf("\nOperation code was changed to %i.\na = %i, b = %i \n", op, a, b);  }    switch (op) {  case 0:  result = a + b;  printf("\nResult: a + b = %.0f\n", result);  break;  case 1:  result = a - b;  printf("\nResult: a - b = %.0f\n", result);  break;  case 2:  result = a \* b;  printf("\nResult: a \* b = %.0f\n", result);  break;  case 3:  if(b != 0) {  result = a / b;  printf("\nResult: a / b = %f\n", result);  } else {  printf("b = 0, a / b can't be computed. \n");  }  break;  case 4:  result = abs(a);  printf("\nResult: |a| = %.0f\n", result);  break;  case 5:  result = fmin(a, b);  printf("\nResult: minimal value is %.0f\n", result);  break;  case 6:  result = fmax(a, b);  printf("\nResult: maximal value is %.0f\n", result);  break;  case 7:  case 13:  if( a == 0 && b < 0) {  printf("Division by zero, result can't be computed \n");  } else {  result = pow(a, b);  printf("\nResult: a^b = %f\n", result);  }  break;  case 8:  result = tan(a\*pi) / (7 - 2\*b);  if(isnan(result) || isinf(result)) {  printf("Result can't be computed \n");  } else {  printf("\nResult: var(a, b) = %f\n", result);  } /\*всегда получаем 0, см. график \*/  break;  default:  if (abs(a+1) != 0 && abs(b+1) != 0) {  result = (op % abs(a+1)) + (op % abs(b+1));  printf("\nResult: def( a, b) = %.0f\n", result);  } else {  printf("Division by zero, result can't be computed \n");  }  break;  }    return 0;  } |
| **#include <stdio.h>**  **#include <math.h>**  **#include <progbase.h>**  **int main(void) {**  **int a = 0;**  **int b = 0;**  **int op = 0;**  **int temp = 0;**  **double result = 0;**  **const double pi = 3.1415926535;**    **printf("List of possible operations:\**  **\n 0\ta + b \n 1\ta - b \n 2\ta \* b \n 3\ta / b \n 4\tabs(a) \n 5\tmin(a, b) \n 6\tmax(a, b) \n 7, 13\tpow(a, b)\**  **\n 8\tvar(a, b) \n other\tdef(a, b)\n");**    **puts("\nEnter operation code:");**  **op = getInt();**  **printf("You entered operation code %i\n", op);**    **puts("\nEnter int a:");**  **a = getInt();**  **printf("You entered: a = %i\n", a);**  **puts("Enter int b:");**  **b = getInt();**  **printf("You entered: b = %i\n", b);**  **if(op < 0){**  **op = abs(op);**    **temp = a;**  **a = b;**  **b = temp;**    **printf("\nOperation code was changed to %i.\na = %i, b = %i \n", op, a, b);**  **}**    **switch (op) {**  **case 0:**  **result = a + b;**  **printf("\nResult: a + b = %.0f\n", result);**  **break;**  **case 1:**  **result = a - b;**  **printf("\nResult: a - b = %.0f\n", result);**  **break;**  **case 2:**  **result = a \* b;**  **printf("\nResult: a \* b = %.0f\n", result);**  **break;**  **case 3:**  **if(b != 0) {**  **result = a / b;**  **printf("\nResult: a / b = %f\n", result);**  **} else {**  **printf("b = 0, a / b can't be computed. \n");**  **}**  **break;**  **case 4:**  **result = abs(a);**  **printf("\nResult: |a| = %.0f\n", result);**  **break;**  **case 5:**  **result = fmin(a, b);**  **printf("\nResult: minimal value is %.0f\n", result);**  **break;**  **case 6:**  **result = fmax(a, b);**  **printf("\nResult: maximal value is %.0f\n", result);**  **break;**  **case 7:**  **case 13:**  **if( a != 0 && b > 0) {**  **result = pow(a, b);**  **printf("\nResult: a^b = %f\n", result);**  **} else{**  **printf("Division by zero, result can't be computed \n");**  **}**  **break;**  **case 8:**  **result = tan(a\*pi) / (7 - 2\*b);**  **if(isnan(result) || isinf(result)) {**  **printf("Result can't be computed \n");**  **} else {**  **printf("\nResult: var(a, b) = %f\n", result);**  **} /\*всегда получаем 0, см. график \*/**  **break;**  **default:**  **result = (op \* abs(a+1)) + (op \* abs(b+1));**  **printf("\nResult: def( a, b) = %.0f\n", result);**  **break;**  **}**    **return 0;**  **}##i#n#c#l#u#d#e# #<#s#t#d#i#o#.#h#>#**  **###i#n#c#l#u#d#e# #<#m#a#t#h#.#h#>#**  **###i#n#c#l#u#d#e# #<#p#r#o#g#b#a#s#e#.#h#>#**  **#**  **#i#n#t# #m#a#i#n#(#v#o#i#d#)# #{#**  **#**  **# #i#n#t# #a# #=# #0#;#**  **# #i#n#t# #b# #=# #0#;#**  **# #i#n#t# #o#p# #=# #0#;#**  **# #i#n#t# #t#e#m#p# #=# #0#;#**  **# #d#o#u#b#l#e# #r#e#s#u#l#t# #=# #0#;#**  **# #c#o#n#s#t# #d#o#u#b#l#e# #p#i# #=# #3#.#1#4#1#5#9#2#6#5#3#5#;#**  **# #**  **# #p#r#i#n#t#f#(#"#L#i#s#t# #o#f# #p#o#s#s#i#b#l#e# #o#p#e#r#a#t#i#o#n#s#:#\#**  **# #\#n# #0#\#t#a# #+# #b# #\#n# #1#\#t#a# #-# #b# #\#n# #2#\#t#a# #\*# #b# #\#n# #3#\#t#a# #/# #b# #\#n# #4#\#t#a#b#s#(#a#)# #\#n# #5#\#t#m#i#n#(#a#,# #b#)# #\#n# #6#\#t#m#a#x#(#a#,# #b#)# #\#n# #7#,# #1#3#\#t#p#o#w#(#a#,# #b#)#\#**  **# #\#n# #8#\#t#v#a#r#(#a#,# #b#)# #\#n# #o#t#h#e#r#\#t#d#e#f#(#a#,# #b#)#\#n#"#)#;# #**  **# #**  **# #p#u#t#s#(#"#\#n#E#n#t#e#r# #o#p#e#r#a#t#i#o#n# #c#o#d#e#:#"#)#;#**  **# #o#p# #=# #g#e#t#I#n#t#(#)#;#**  **# #p#r#i#n#t#f#(#"#Y#o#u# #e#n#t#e#r#e#d# #o#p#e#r#a#t#i#o#n# #c#o#d#e# #%#i#\#n#"#,# #o#p#)#;#**  **# #**  **# #p#u#t#s#(#"#\#n#E#n#t#e#r# #i#n#t# #a#:#"#)#;#**  **# #a# #=# #g#e#t#I#n#t#(#)#;#**  **# #p#r#i#n#t#f#(#"#Y#o#u# #e#n#t#e#r#e#d#:# #a# #=# #%#i#\#n#"#,# #a#)#;#**  **#**  **# #p#u#t#s#(#"#E#n#t#e#r# #i#n#t# #b#:#"#)#;#**  **# #b# #=# #g#e#t#I#n#t#(#)#;#**  **# #p#r#i#n#t#f#(#"#Y#o#u# #e#n#t#e#r#e#d#:# #b# #=# #%#i#\#n#"#,# #b#)#;#**  **#**  **#**  **# #i#f#(#o#p# #<# #0#)#{#**  **# # #o#p# #=# #a#b#s#(#o#p#)#;#**  **# # #**  **# # #t#e#m#p# #=# #a#;#**  **# # #a# #=# #b#;#**  **# # #b# #=# #t#e#m#p#;#**  **# # #**  **# # #p#r#i#n#t#f#(#"#\#n#O#p#e#r#a#t#i#o#n# #c#o#d#e# #w#a#s# #c#h#a#n#g#e#d# #t#o# #%#i#.#\#n#a# #=# #%#i#,# #b# #=# #%#i# #\#n#"#,# #o#p#,# #a#,# #b#)#;#**  **# # #}#**  **# # # #**  **# #s#w#i#t#c#h# #(#o#p#)# #{#**  **# # # # # #c#a#s#e# #0#:#**  **# # # # # # #r#e#s#u#l#t# #=# #a# #+# #b#;#**  **# # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #a# #+# #b# #=# #%#.#0#f#\#n#"#,# #r#e#s#u#l#t#)#;#**  **# # #b#r#e#a#k#;#**  **# # # # # #c#a#s#e# #1#:#**  **# # # # # # #r#e#s#u#l#t# #=# #a# #-# #b#;#**  **# # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #a# #-# #b# #=# #%#.#0#f#\#n#"#,# #r#e#s#u#l#t#)#;#**  **# # #b#r#e#a#k#;#**  **# # # # # #c#a#s#e# #2#:#**  **# # #r#e#s#u#l#t# #=# #a# #\*# #b#;#**  **# # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #a# #\*# #b# #=# #%#.#0#f#\#n#"#,# #r#e#s#u#l#t#)#;#**  **# # #b#r#e#a#k#;#**  **# # # # # #c#a#s#e# #3#:#**  **# # # # # # #i#f#(#b# #!#=# #0#)# #{#**  **# # # # # # # #r#e#s#u#l#t# #=# #a# #/# #b#;#**  **# # # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #a# #/# #b# #=# #%#f#\#n#"#,# #r#e#s#u#l#t#)#;#**  **# # # #}# #e#l#s#e# #{#**  **# # # #p#r#i#n#t#f#(#"#b# #=# #0#,# #a# #/# #b# #c#a#n#'#t# #b#e# #c#o#m#p#u#t#e#d#.# #\#n#"#)#;#**  **# # # #}#**  **# # #b#r#e#a#k#;#**  **# # # # # #c#a#s#e# #4#:#**  **# # # # # # #r#e#s#u#l#t# #=# #a#b#s#(#a#)#;#**  **# # # # # # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #|#a#|# #=# #%#.#0#f#\#n#"#,# #r#e#s#u#l#t#)#;#**  **# # # # # # #b#r#e#a#k#;#**  **# # # # # #c#a#s#e# #5#:#**  **# # # # # # #r#e#s#u#l#t# #=# #f#m#i#n#(#a#,# #b#)#;#**  **# # # # # # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #m#i#n#i#m#a#l# #v#a#l#u#e# #i#s# #%#.#0#f#\#n#"#,# #r#e#s#u#l#t#)#;# # # # # # #**  **# # # # # # #b#r#e#a#k#;#**  **# # # # # #c#a#s#e# #6#:#**  **# # #r#e#s#u#l#t# #=# #f#m#a#x#(#a#,# #b#)#;#**  **# # # # # # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #m#a#x#i#m#a#l# #v#a#l#u#e# #i#s# #%#.#0#f#\#n#"#,# #r#e#s#u#l#t#)#;# # # # # # #**  **# # # # # # #b#r#e#a#k#;#**  **# # # # # #c#a#s#e# #7#:#**  **# # # # # #c#a#s#e# #1#3#:#**  **# # # # # # #i#f#(# #a# #=#=# #0# #&#&# #b# #<# #0#)# #{# #**  **# # # #p#r#i#n#t#f#(#"#D#i#v#i#s#i#o#n# #b#y# #z#e#r#o#,# #r#e#s#u#l#t# #c#a#n#'#t# #b#e# #c#o#m#p#u#t#e#d# #\#n#"#)#;#**  **# # # #}# #e#l#s#e# #{#**  **# # # # # # # #r#e#s#u#l#t# #=# #p#o#w#(#a#,# #b#)#;#**  **# # # # # # # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #a#^#b# #=# #%#f#\#n#"#,# #r#e#s#u#l#t#)#;#**  **# # # # # # # #}# #**  **# # # # # # #b#r#e#a#k#;#**  **# # # # # #c#a#s#e# #8#:#**  **# # # # # # #r#e#s#u#l#t# #=# #t#a#n#(#a#\*#p#i#)# #/# #(#7# #-# #2#\*#b#)#;#**  **# # # # # # #i#f#(#i#s#n#a#n#(#r#e#s#u#l#t#)# #|#|# #i#s#i#n#f#(#r#e#s#u#l#t#)#)# #{#**  **# # # #p#r#i#n#t#f#(#"#R#e#s#u#l#t# #c#a#n#'#t# #b#e# #c#o#m#p#u#t#e#d# #\#n#"#)#;#**  **# # # # # # # #}# #e#l#s#e# #{#**  **# # # # # # # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #v#a#r#(#a#,# #b#)# #=# #%#f#\#n#"#,# #r#e#s#u#l#t#)#;#**  **# # # # # # # #}# #/#\*#2#A#5#3#4#0# #?#>#;#C#G#0#5#<# #0#,# #A#<#.# #3#@#0#D#8#:# #\*#/#**  **# # # # # # #b#r#e#a#k#;#**  **# # # # # #d#e#f#a#u#l#t#:#**  **# # # # # # #i#f# #(#a#b#s#(#a#+#1#)# #!#=# #0# #&#&# #a#b#s#(#b#+#1#)# #!#=# #0#)# #{#**  **# # # # # # # #r#e#s#u#l#t# #=# #(#o#p# #%# #a#b#s#(#a#+#1#)#)# #+# #(#o#p# #%# #a#b#s#(#b#+#1#)#)#;#**  **# # # #p#r#i#n#t#f#(#"#\#n#R#e#s#u#l#t#:# #d#e#f#(# #a#,# #b#)# #=# #%#.#0#f#\#n#"#,# #r#e#s#u#l#t#)#;#**  **# # # #}# #e#l#s#e# #{#**  **# # # #p#r#i#n#t#f#(#"#D#i#v#i#s#i#o#n# #b#y# #z#e#r#o#,# #r#e#s#u#l#t# #c#a#n#'#t# #b#e# #c#o#m#p#u#t#e#d# #\#n#"#)#;#**  **# # # #}# #**  **# # #b#r#e#a#k#;#**  **# # # # # #}#**  **# # # # # #**  **# # # # #r#e#t#u#r#n# #0#;#**  **#}#** |

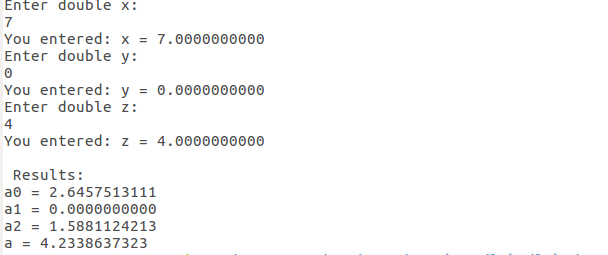
**Завдання 4.**

|  |
| --- |
| main.c |
| **#include <stdio.h>**  **#include <math.h>**  **#include <progbase.h>**  **int main(void) {**  **const double pi = 3.1415926535;**  **unsigned int n = 0;**  **unsigned int m = 0;**  **int i = 1;**  **int j = 1;**  **double x = 0;**  **double sumj = 0;**      **puts("Enter unsigned int n:");**  **n = getInt();**  **printf("You entered: n = %i\n", n);**  **puts("Enter unsigned int m:");**  **m = getInt();**  **printf("You entered: m = %i\n", m);**    **for(; j <= m; j++){**  **sumj += cos((pi\*j)/2);**  **}**  **for(; i <= n; i++) {**  **x +=(i+1) + sumj;**  **}**    **printf("x = %f\n", x);**  **return 0;**  **}** |

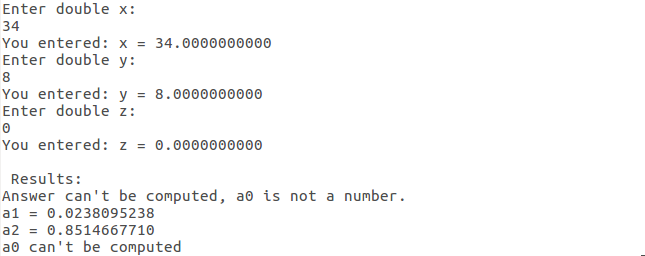
**Приклади результатів**

**Завдання 1.**

Результат роботи програми у випадку, коли введені значення не викликають помилок.



Робота програми у випадку, коли виникає помилка з обчисленням одного з значень.

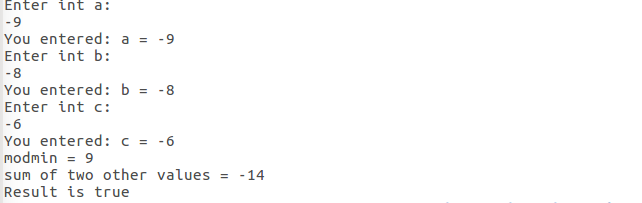


**Завдання 2.**

Введено два додатних числа та 0, результат роботи програми true.



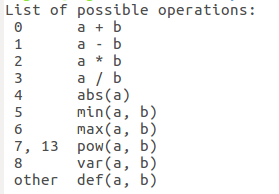
Введені від’ємні значення, в результаті виводятся значення modmin і sum2, а результат true.



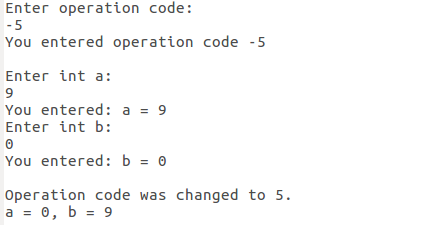
 Результатом роботи програми буде значення false.

**Завдання 3.**

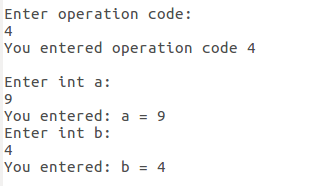
При запуску програми виводитиметься список доступних операцій та їх коди.



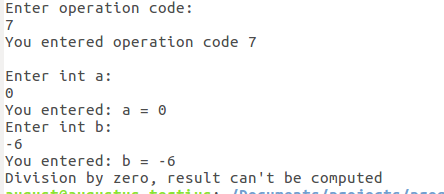
При введенні від’ємного коду операції він замінюватиметься на відповідний додатний а значення a і b, введедні користувачем, змінюватимуться місцями, про що виводитиметься повідомлення.



Приклад роботи програми, яка не викликає обробку помилок.

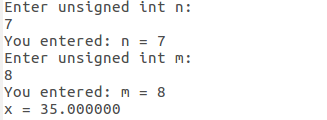


В такій ситуації програма виводить повідомлення про недопустиму операцію.



**Завдання 4.**

При вводі значень m і n отримаємо значення х.



**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.

В завданні 4 була відпрацьована робота з циклічними конструкціями, за допомогою яких були виконані обчислення на основі даних, введених користувачем.

Компіляція всього коду відбувалася за допомогою компілятора gcc.   
Було відпрацьовано навички роботи з компілятором, а також його додатковими опціями.