

Лабораторна робота №5. Циклічні конструкції.

1 Вимоги

1.1 Розробник

- Журавльов Ярослав Юрійович;
- студент групи КІТ-120а;
- 09-ноя-2020

1.2 Загальне завдання

Розробити програми, вирішення яких потребує використання циклічних конструкцій.

1.3 Індивідуальне завдання

Завдання 4. Розробити програму, яка визначає чи є задане число досконалим.

2. Опис роботи

2.1 Функціональне призначення

Програма призначена для визначення чи є задане число досконалим.

Результат зберігається у змінній *is_perfect_number*.

Демонстрація знайденого результату передбачає покрокове виконання програми.

2.2 Опис логічної структури

Якщо задане число менше або дорівнює 1, то число не є досконалим.

За допомогою циклу перебираємо усі можливі числа у діапазоні від 2 до заданого числа, поділеного на 2, включно. Якщо число є дільником заданого числа, то додаємо цей дільник до змінної *divider_sum_of_number*. Якщо сума дільників числа дорівнює числу, то число є досконалим, змінюємо значення змінної *is_perfect_number* на 1.

2.3 Важливі фрагменти програми

2.3.1 Вирішення задачі з допомогою циклу for.

2.3.1.1 Число 28, яке є досконалим

Перебір усіх чисел, які можуть бути дільниками числа

```
for (int divider = 2; divider <= NUMBER_TASK_4 / 2; divider++)
```

Перевірка, чи є число дільником заданого числа та додавання дільників до змінної, яка містить суму дільників. Дільниками є 1, 2, 4, 7, 14

```
if (NUMBER_TASK_4 % divider == 0) divider_sum_of_number += divider;
```

Перевірка чи дорівнює сума дільників заданому числу та чи може задане число бути досконалим

```
if (divider_sum_of_number == NUMBER_TASK_4 && NUMBER_TASK_4 > 1) is_perfect_number = 1;
```

Перевірка результату

```
(gdb) print(is_perfect_number)
$1 = 1
```

2.3.1.2 Число 10, яке не досконалим

Перебір усіх чисел, які можуть бути дільниками числа

```
for (int divider = 2; divider <= NUMBER_TASK_4 / 2; divider++)
```

Перевірка, чи є число дільником заданого числа та додавання дільників до змінної, яка містить суму дільників. Дільниками є 1, 2, 5

```
if (NUMBER_TASK_4 % divider == 0) divider_sum_of_number += divider;
```

Перевірка чи дорівнює сума дільників заданому числу та чи може задане число бути досконалим

```
if (divider_sum_of_number == NUMBER_TASK_4 && NUMBER_TASK_4 > 1) is_perfect_number = 1;
```

Перевірка результату

```
(gdb) print(is_perfect_number)
$1 = 0
```

2.3.2 Вирішення задачі з допомогою циклу while.

2.3.2.1 Число 28, яке є досконалим

Перебір усіх чисел, які можуть бути дільниками числа, зберігаються у змінній *divider*

```
int divider = 2;
while (divider <= NUMBER_TASK_4 / 2)
```

Перевірка, чи є число дільником заданого числа та додавання дільників до змінної, яка містить суму дільників

```
if (NUMBER_TASK_4 % divider == 0) divider_sum_of_number += divider;
```

Збільшення змінної *divider*

```
divider++;
```

Перевірка чи дорівнює сума дільників заданому числу та чи може задане число бути досконалим

```
if (divider_sum_of_number == NUMBER_TASK_4 && NUMBER_TASK_4 > 1) is_perfect_number = 1;
```

Перевірка результату

```
(gdb) print(is_perfect_number)
$1 = 1
```

2.3.2.2 Число 10, яке не досконалим

Перебір усіх чисел, які можуть бути дільниками числа

```
int divider = 2;
while (divider <= NUMBER_TASK_4 / 2)
```

Перевірка, чи є число дільником заданого числа та додавання дільників до змінної, яка містить суму дільників

```
if (NUMBER_TASK_4 % divider == 0) divider_sum_of_number += divider;
```

Збільшення змінної *divider*

```
divider++;
```

Перевірка чи дорівнює сума дільників заданому числу та чи може задане число бути досконалим

```
if (divider_sum_of_number == NUMBER_TASK_4 && NUMBER_TASK_4 > 1) is_perfect_number = 1;
```

Перевірка результату

```
(gdb) print(is_perfect_number)
$1 = 0
```

```
(gdb) print(is_perfect_number)
$1 = 0
```

2.3.3 Вирішення задачі з допомогою циклу do while.

2.3.3.1 Число 28, яке є досконалим

Перевірка чи може задане число бути досконалим

```
if (NUMBER_TASK_4 > 1){
```

Перебір усіх чисел, які можуть бути дільниками числа, зберігаються у змінній *divider*

```
do {  
    if (NUMBER_TASK_4 % divider == 0) divider_sum_of_number += divider;  
    divider++;  
} while (divider <= NUMBER_TASK_4 / 2);
```

Перевірка, чи є число дільником заданого числа та додавання дільників до змінної, яка містить суму дільників

```
if (NUMBER_TASK_4 % divider == 0) divider_sum_of_number += divider;
```

Збільшення змінної *divider*

```
divider++;
```

Перевірка чи дорівнює сума дільників заданому числу та чи може задане число бути досконалим

```
if (divider_sum_of_number == NUMBER_TASK_4 && NUMBER_TASK_4 > 1) is_perfect_number = 1;
```

Перевірка результату

```
(gdb) print(is_perfect_number)  
$1 = 1
```

2.3.3.2 Число 10, яке не досконалим

Перевірка чи може задане число бути досконалим

```
if (NUMBER_TASK_4 > 1){
```

Перебір усіх чисел, які можуть бути дільниками числа

```
do {  
    if (NUMBER_TASK_4 % divider == 0) divider_sum_of_number += divider;  
    divider++;  
} while (divider <= NUMBER_TASK_4 / 2);
```

Перевірка, чи є число дільником заданого числа та додавання дільників до змінної, яка містить суму дільників

```
if (NUMBER_TASK_4 % divider == 0) divider_sum_of_number += divider;
```

Збільшення змінної *divider*

```
divider++;
```

Перевірка чи дорівнює сума дільників заданому числу та чи може задане число бути досконалим

```
if (divider_sum_of_number == NUMBER_TASK_4 && NUMBER_TASK_4 > 1) is_perfect_number = 1;
```

Перевірка результату

```
(gdb) print(is_perfect_number)
$1 = 0
```

Висновки:

У результаті виконання лабораторної роботи отримали навички використання циклічних конструкцій для вирішення задач.