

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження рекурсивних
алгоритмів»

Варіант 15

Виконав студент ІІ-12, Кириченко Владислав Сергійович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Лабораторна робота № 6

Назва роботи: Дослідження рекурсивних алгоритмів

Мета: дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Варіант 15

Умова задачі:

Обчислити добуток елементів арифметичної прогресії, що зростає: початкове значення – 2, кінцеве – 30, крок – 4

Постановка задачі:

Нам дані значення першого та останнього елемента арифметичної прогресії, та значення кроку. Результатом буде значення добутку усіх елементів цієї прогресії.

Побудова математичної моделі:

Складемо таблицю змінних:

Змінна	Тип	Ім'я	Призначення
перший елемент арифметичної прогресії	цілочисельний	<i>a</i>	Початкові дані
останній елемент арифметичної прогресії	цілочисельний	<i>aEnd</i>	Початкові дані
крок арифметичної прогресії	цілочисельний	<i>d</i>	Початкові дані
формальний параметр(перший елемент арифметичної прогресії)	цілочисельний	<i>firdstEl</i>	Проміжкове значення
формальний параметр(останній елемент арифметичної прогресії)	цілочисельний	<i>lastEl</i>	Проміжкове значення
формальний параметр(крок арифметичної прогресії)	цілочисельний	<i>step</i>	Проміжкове значення
результат роботи програми	цілочисельний	<i>r</i>	Результат
результат роботи підпрограми	цілочисельний	<i>rFunc</i>	Результат(підпрограма)

3. Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізація отримання значення добутку усіх елементів цієї прогресії(**r**)

Псевдокод(основна програма):

Крок 1.

початок

введення *a, aEnd, d*

знаходження значення змінної **r**

виведення **r**

кінець

Крок 2.

початок

введення *a, aEnd, d*

r = seqProduct(a, aEnd, d)

виведення **r**

кінець

Псевдокод(підпрограма):

Крок 1.

функція seqProduct(firdstEl, lastEl, step)

початок

введення *firdstEl, lastEl, step*

перевірка чи **firdstEl == lastEl**

обрахування **rFunc**

повернути **r**

кінець

Крок 2.

функція seqProduct(*firdstEl*,*lastEl*,*step*)

початок

введення *firdstEl*,*lastEl*,*step*

якщо *firdstEl*==*lastEl*

ТО

повернути *firdstEl*

все якщо

обрахування *rFunc*

повернути *rFunc*

кінець

Крок 3.

функція seqProduct(*firdstEl*,*lastEl*,*step*)

початок

введення *firdstEl*,*lastEl*,*step*

якщо *firdstEl*==*lastEl*

ТО

повернути *firdstEl*

все якщо

rFunc= *firdstEl**seqProduct(*firdstEl*+*step*,*lastEl*,*step*)

повернути *rFunc*

кінець

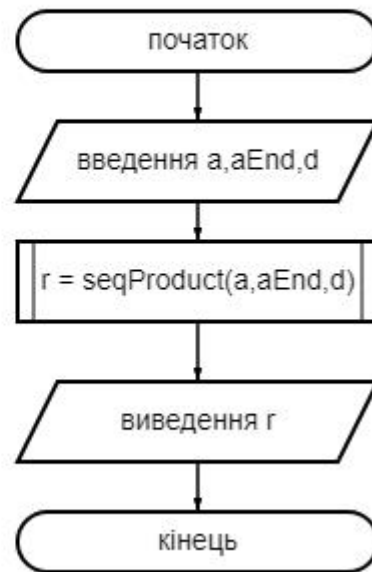
Блок схема:

Основна програма

Крок 1

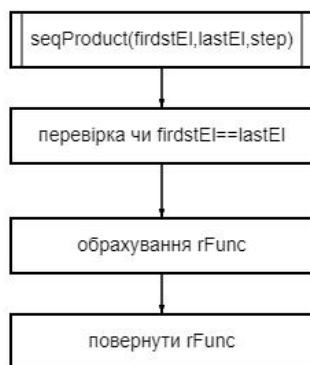


Крок 2



Підпрограма

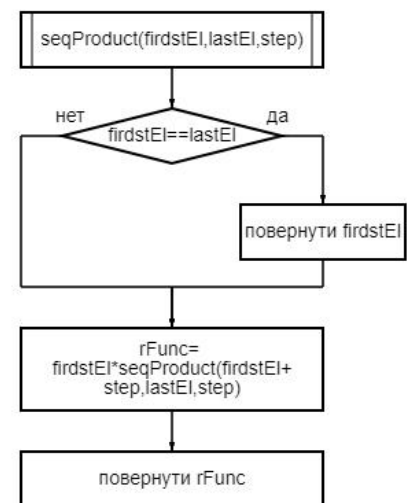
Крок 1



Крок 2



Крок 3



4. Код програми(C++)

The screenshot shows the Visual Studio Code editor with a C++ file named `lab6.cpp` open. The code defines a recursive function `seqProduct` and a `main` function. The `main` function initializes `a=2`, `aEnd=30`, and `d=4`, then calls `seqProduct(a, aEnd, d)`. The output in the Command Prompt shows the values of `a`, `aEnd`, `d`, and the final result of the recursive calculation, which is 518918400.

```
lab6 > g++ lab6.cpp
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int seqProduct(int a,int aEnd, int d);
6
7 int main() {
8     int a=2;
9     int aEnd=30;
10    int d=4;
11    int r;
12    r = seqProduct(a,aEnd,d);
13
14    cout << "a = " <<setw(29)<< a << endl
15         << "aEnd ="<<setw(28)<< aEnd << endl
16         << "d = " <<setw(29)<< d << endl
17         << "Добуток усіх членів прогресії = "<< r ;
18    return 0;
19 }
20
21 int seqProduct(int firdstEl, int lastEl, int step) {
22     int rFunc;
23     if(firdstEl==lastEl)
24         return firdstEl;
25     rFunc= firdstEl*seqProduct(firdstEl+step,lastEl,step);
26     return rFunc;
27 }
```

Command Prompt Output:

```
E:\files\kpi\asd\lab6>a
a = 2
aEnd = 30
d = 4
Добуток усіх членів прогресії = 518918400
E:\files\kpi\asd\lab6>
```

5.Перевірка алгоритму

Блок	Дія
	Початок
1	Введення $a=2, aEnd=30, step=4$
2	початок дії підпрограми <code>seqProduct(firdstEl=2,lastEl=30,step=4)</code>
3	$firdstEl==lastEl$ - false
4	збільшення стеку $rFunc = firdstEl * seqProduct(6,30,4);$
5	$firdstEl==lastEl$ - false
6	збільшення стеку $rFunc = firdstEl * seqProduct(10,30,4);$

7	<i>firdstEl==lastEl - false</i>
8	збільшення стеку <i>rFunc= firdstEl*seqProduct(14,30,4);</i>
9	<i>firdstEl==lastEl - false</i>
10	збільшення стеку <i>rFunc= firdstEl*seqProduct(18,30,4);</i>
11	<i>firdstEl==lastEl - false</i>
12	збільшення стеку <i>rFunc= firdstEl*seqProduct(22,30,4);</i>
13	<i>firdstEl==lastEl - false</i>
14	збільшення стеку <i>rFunc= firdstEl*seqProduct(26,30,4);</i>
15	<i>firdstEl==lastEl - false</i>
16	збільшення стеку <i>rFunc= firdstEl*seqProduct(30,30,4);</i>
17	<i>firdstEl==lastEl - true</i>
18	повернути 30
19	зменшення стеку
20	повернути rFunc=780
21	зменшення стеку
22	повернути rFunc=17160

23	<i>зменшення стеку</i>
24	<i>повернути rFunc=308880</i>
25	<i>зменшення стеку</i>
26	<i>повернути rFunc=4324320</i>
27	<i>зменшення стеку</i>
28	<i>повернути rFunc=43243200</i>
29	<i>зменшення стеку</i>
30	<i>повернути rFunc=259459200</i>
31	<i>зменшення стеку</i>
32	<i>повернути rFunc=518918400</i>
33	<i>кінець виконання підпрограми</i>
	<i>виведення 518918400</i>

Висновок - Було досліджено особливості роботи рекурсивних алгоритмів та набуто практичних навичок їх використання під час складання програмних специфікацій підпрограм.