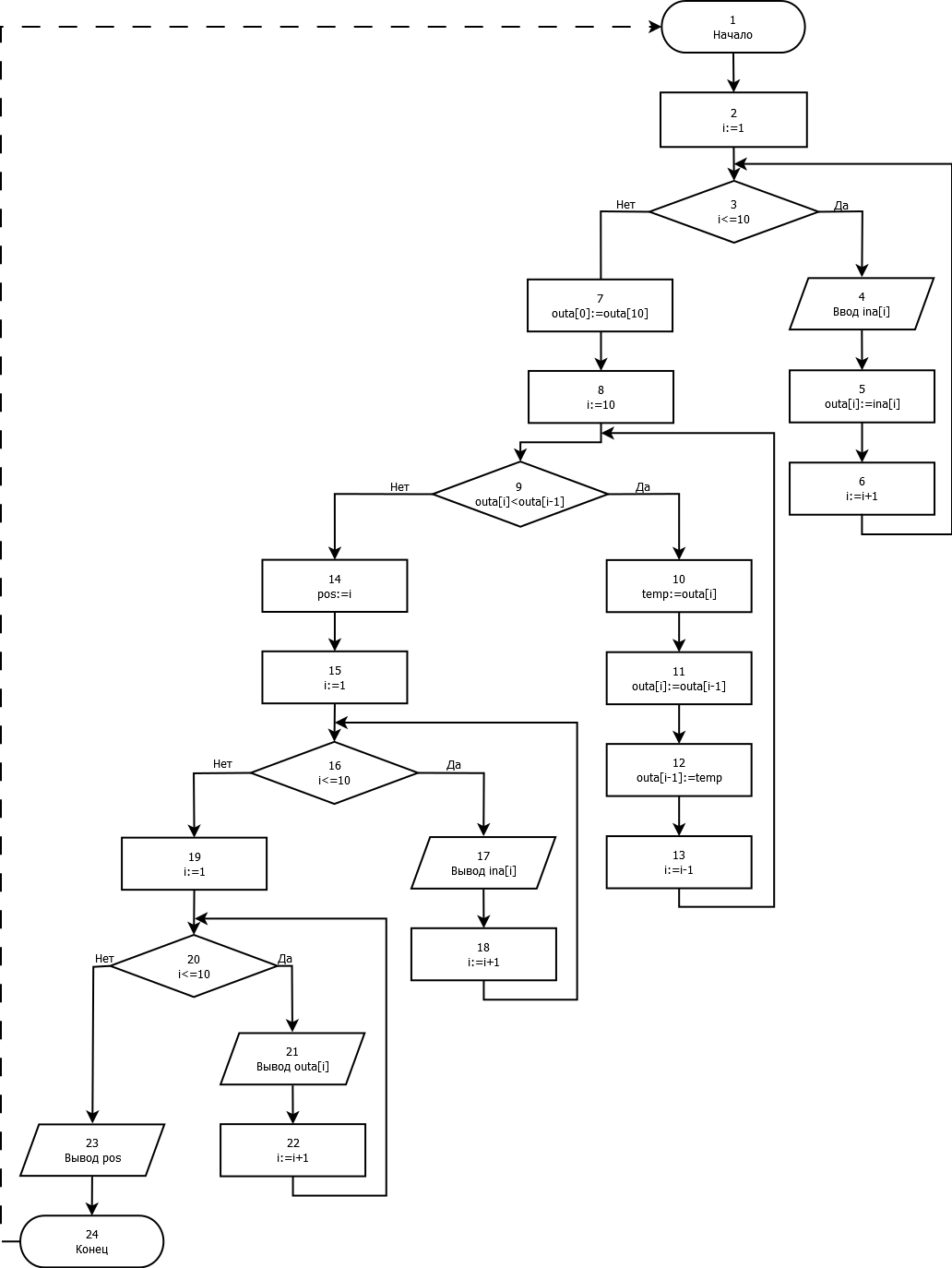
**Индивидуальное задание № 7. Условие:**

Ввести массив А (10), первые девять элементов которого упорядочены по возрастанию. Поместить последний элемент массива в соответствующее место массива, чтобы не нарушить его упорядоченность. Вывести исходный и результирующий массивы и номер помещенного элемента.

**Схема алгоритма:**



**Описание схемы алгоритма:**

Назначение переменных:

**i** – переменная целого типа, используется как счетчик циклов и вывода значений массивов на экран;

**temp** – переменная целого типа, используется для хранения значения элемента массива при обмене местами двух элементов массива;

**pos** – переменная целого типа, используется для хранения позиции, в которую был помещен последний элемент для полного упорядочивания массива;

**ina[1..10]** – массив целых чисел из 10 элементов, в котором хранятся введенные значения (неупорядоченный массив);

**outa[0..10]** – массив целых чисел из 11 элементов, в котором хранятся значения из массива **ina[1..10]**, упорядоченные по возрастанию;

Назначение блоков алгоритма:

1 – начало программы;

2 – установка начального значения счетчика цикла;

3 – условие цикла для ввода значений элементов массива;

4 – ввод значения очередного элемента массива ina;

5 – установка значения очередного элемента массива outa;

6 – изменение счетчика цикла;

7 – установка значения первого элемента массива outa для предотвращения выхода за пределы массива;

8 – установка начального значения счетчика цикла;

9 – условие для проверки упорядоченности массива outa;

10 – установка значения переменной temp, равного значению i-го элемента массива outa;

11 – установка значения i-го элемента массива outa, равного значению i-1-го элемента массива outa;

12 – установка значения i-1-го элемента массива outa, равного значению переменной temp;

13 – изменение счетчика цикла;

14 – установка значения переменной pos, равного позиции, в которую был перемещен последний элемент массива outa;

15 – установка начального значения счетчика цикла;

16 – условие цикла для вывода значения элемента массива ina;

17 – вывод значения очередного элемента массива ina;

18 – изменение счетчика цикла;

19 – установка начального значения счетчика цикла;

20 – условие цикла для вывода значения элемента массива outa;

21 – вывод значения очередного элемента массива outa;

22 – изменение счетчика цикла;

23 – вывод значения переменной pos;

24 – конец программы.

**Расчет метрики Маккейба для алгоритма и определение базисных независимых путей:**

Цикломатическое число Маккейба равно

**Z(G) = e – v + 2p = 27 – 24+ 2 = 5**

Где **e** – число дуг ориентированного графа **G**; **v** – число вершин; **р** – число компонентов связности графа.

Базисные независимые пути в алгоритме:

Первый вариант:

1. 1-2-3(нет)-7-8-9(нет)-14-15-16(нет)-19-20(нет)-23-24
2. 1-2-3(да)-4-5-6-3(нет)-7-8-9(нет)-14-15-16(нет)-19-20(нет)-23-24
3. 1-2-3(нет)-7-8-9(да)-10-11-12-13-9(нет)-14-15-16(нет)-19-20(нет)-23-24
4. 1-2-3(нет)-7-8-9(нет)-14-15-16(да)-17-18-16(нет)-19-20(нет)-23-24
5. 1-2-3(нет)-7-8-9(нет)-14-15-16(нет)-19-20(да)-21-22-20(нет)-23-24

Второй вариант:

1. 1-2-3(нет)-7-8-9(нет)-14-15-16(нет)-19-20(нет)-23-24
2. 1-2-3(да)-4-5-6-3(нет)-7-8-9(нет)-14-15-16(нет)-19-20(нет)-23-24
3. 1-2-3(да)-4-5-6-3(нет)-7-8-9(да)-10-11-12-13-9(нет)-14-15-16(нет)-19-20(нет)-23-24
4. 1-2-3(да)-4-5-6-3(нет)-7-8-9(нет)-14-15-16(да)-17-18-16(нет)-19-20(нет)-23-24
5. 1-2-3(нет)-7-8-9(нет)-14-15-16(нет)-19-20(да)-21-22-20(нет)-23-24

**Расчет метрики Джилба для алгоритма:**

***CL* = 4**,

**cl = 0,182** (количество операторов программы равно 22),

***CLI = 0***.

Где **CL** — количество условных и циклических операторов, **cl** — насыщенность программы условными и циклическими операторами, **CLI** — максимальный уровень вложенности условного и циклического оператора.

**Расчет метрики граничных значений для алгоритма:**

**Свойства подграфов программы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Характеристики**  **подграфов программ** | **Номер вершины выбора** | | | |
| **3** | **9** | **16** | **20** |
| Вершины перехода | 4, 7 | 10, 14 | 17, 19 | 21, 23 |
| Вершины подграфа | 3, 4, 5, 6 | 9, 10,  11, 12,13 | 16, 17, 18 | 20, 21, 22 |
| Нижняя граница подграфа | 7 | 14 | 19 | 23 |
| Скорректированная сложность вершины графа | 5 | 6 | 4 | 4 |

**Скорректированные сложности вершин графа программы**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины графа программы | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Скорректированная сложность вершины графа | 1 | 1 | 5 | 1 | 1 | 1 | 1 | 1 | 6 | 1 | 1 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины графа программы | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | Sa |
| Скорректированная сложность вершины графа | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 0 | 38 |

**Sa** – абсолютная граничная сложность программы.

**S0** – относительная граничная сложность программы,

**v** – общее число вершин графа программы.

**S0=1-(v-1)/Sa,**

Таким образом, относительная сложность программы равна:

**S0 = 1 - (24-1)/38 = 0,395.**

**Метрики сложности потока управления программ:**

|  |  |
| --- | --- |
| Метрики сложности потока управления | Значения |
|
| Метрика Маккейба *Z(G)* | 5 |
| Абсолютная сложность программы *CL* по метрике Джилба | 4 |
| Относительная сложность программы *cl* по метрике Джилба | 0,182 |
| Максимальный уровень вложенности условного оператора *CLI* по метрике Джилба | 0 |
| Метрика граничных значений (абсолютная граничная сложность программы) *Sa* | 38 |
| Метрика граничных значений (относительная граничная сложность программы) *S0* | 0.395 |