

Лабораторная работа 5

5.1 [#30]

Проведите сборку программы IFSort3 из приведенных ниже разделов проекта.

Для каждого этапа сборки предоставьте тестовые данные такие, чтобы можно было протестировать все альтернативы.

Выполнение:

```
INPUT: cab
OUTPUT: Входные данные cab
        Сортированные данные abc
INPUT: bca
OUTPUT: Входные данные bca
        Сортированные данные abc
```

Разделы проекта.

DP1

```
PROGRAM IFSort3(INPUT, OUTPUT);
{Сортирует 3-строку из INPUT в OUTPUT}
VAR
    Ch1, Ch2, Ch3:Char;
BEGIN {IFSort3}
    READ(Ch1, Ch2, Ch3);
    WRITELN('Входные данные ', Ch1, Ch2, Ch3);
    WRITE('Сортированные данные');
    {Сортируем Ch1, Ch2, Ch3 в OUTPUT}
END. {IFsort3}
```

DP2.1

```
BEGIN {сортируем Ch1, Ch2, Ch3 в OUTPUT }
    IF Ch1 < Ch2
    THEN
        {Ch1 < Ch2:сортируем Ch1, Ch2, Ch3 в OUTPUT}
    ELSE
        {Ch2 <= Ch1:сортируем Ch1, Ch2, Ch3 в OUTPUT}
    END
```

DP 2.1.1

```
{Ch1 < Ch2:сортируем Ch1, Ch2, Ch3 в OUTPUT}
IF Ch2 < Ch3
THEN {Ch1 < Ch2 < Ch3:сортируем Ch1, Ch2, Ch3 в OUTPUT}
    WRITELN(Ch1, Ch2, Ch3)
ELSE
    {Ch1 < Ch2, Ch3 <= Ch2:сортируем Ch1, Ch2, Ch3 в OUTPUT}
```

DP 2.1.2

```
{Ch2 <= Ch1:сортируем Ch1, Ch2, Ch3 в OUTPUT}
IF Ch1 < Ch3
THEN {Ch2 <= Ch1 < Ch3:сортируем Ch1, Ch2, Ch3 в OUTPUT}
    WRITELN(Ch2, Ch1, Ch3)
ELSE
    {Ch2 <= Ch1, Ch3 <= Ch1:сортируем Ch1, Ch2, Ch3 в OUTPUT}
```

```

.....
DP 2.1.2.1
    {Ch2 <= Ch1, Ch3 <= Ch1:сортируем Ch1, Ch2, Ch3 в OUTPUT}
    IF Ch2 < Ch3
    THEN {Ch2 < Ch3 <= Ch1:сортируем Ch1, Ch2, Ch3 в OUTPUT}
        WRITELN(Ch2, Ch3, Ch1)
    ELSE {Ch3 <= Ch2 <= Ch1:сортируем Ch1, Ch2, Ch3 в OUTPUT}
        WRITELN(Ch3, Ch2, Ch1)
.....
DP 2.1.1.1
    {Ch1, Ch2, Ch3 <= Ch2:сортируем Ch1, Ch2, Ch3 в OUTPUT}
    IF Ch1 < Ch3
    THEN {Ch1 < Ch3 <= Ch2:сортируем Ch1, Ch2, Ch3 в OUTPUT}
        WRITELN(Ch1, Ch3, Ch2)
    ELSE {Ch3 <= Ch1 < Ch2:сортируем Ch1, Ch2, Ch3 в OUTPUT}
        WRITELN(Ch3, Ch1, Ch2)

```

5.2 [#30]

Проведите сборку программы MinSort3 из приведенных ниже разделов проекта. Для каждого этапа сборки предоставьте тестовые данные такие, чтобы можно было протестировать все альтернативы.

```

DP 3
PROGRAM MinSort3 (INPUT,OUTPUT);
{сортирует 3-строку из INPUT в OUTPUT }
VAR Ch1, Ch2, Ch3: CHAR;
BEGIN {MinSort3}
    READ(Ch1, Ch2, Ch3);
    WRITELN('Входные данные ', Ch1, Ch2, Ch3);
    WRITE('Сортированные данные ');
    {Печатать минимум в OUTPUT, сохранить содержимое в Ch1 and Ch2
};
    { Сортировать Ch1, Ch2 в OUTPUT };
    WRITELN
END.{Minsort3}
.....
DP 3.1
BEGIN {Печатать минимум в OUTPUT, сохранить содержимое в Ch1 and
Ch2 };
    IF Ch1 < Ch2
    THEN
        { Печатать минимум из Ch1, Ch3 в OUTPUT,
        переместить Ch3 в Ch1,если необходимо}
        IF Ch1 < Ch3
        THEN
            BEGIN
                WRITE(Ch1);
                Ch1 := Ch3
            END
        ELSE
            WRITE(Ch3)
    ELSE
        { Печатать минимум из Ch2, Ch3 в OUTPUT,
        переместить Ch3 в Ch2,если необходимо}

```

```
IF Ch2 < Ch3
THEN
  BEGIN
    WRITE (Ch2);
    Ch2 := Ch3
  END
ELSE
  WRITE (Ch3)
END
.....
DP 3.1
BEGIN {Сортируем Ch1, Ch2 в OUTPUT }
  IF Ch1 < Ch2
  THEN
    WRITE (Ch1, Ch2)
  ELSE
    WRITE (Ch2, Ch1)
END
```

5.3 [#10]

Внесите изменения в программу IFSort3, чтобы сортировка выполнялась в формате:

```
INPUT: acb
OUTPUT: Входные данные (a,c,b) сортируются в (a,b,c).
```