Задания для самостоятельной работы обучающегося и методические рекомендации по их выполнению

СРО №14, 15. Ссылочный тип. Динамические переменные

Задания:

1. Привести примеры всех перечисленных в лекции по Теме "Ссылочный тип. Динамические переменные" действий и операций, связанных со ссылочным типом. Отдельно рассмотреть типизированные и нетипизированные указатели. В качестве указываемых объектов использовать обычные статические переменные.
2. Описать переменные: указатель на массив, массив указателей, указатель на запись, запись с полем-указателем. После этого выпишите, как организовать доступ к элементам этих структур при использовании разыменования "^".
3. В программе описаны типы и переменные:

```
type
  TA = array[1..5] of Double;
  TB = array[1..20] of LongInt;
  TC = array[1..30] of Integer;
  TD = array[1..20] of Boolean;
var
  A: ^TA;  B: ^TB;  C: ^TC;  D: ^TD;
  E: TB;
```

   После выполнения инструкций присваивания массивы A^, B^, C^, D^ будут "наложены" на какие-то участки единственного реального массива E:

```
B := @E; A := @B^[11]; C := @B^[6]; D := @B^[16];
```

   Определить смещения в байтах элементов массивов A^[4], B^[10], C^[20], D^[5] от начала массива E. Все ли массивы A^, B^, C^, D^ целиком "покрываются" массивом E?

4. Запрограммируйте фрагмент из лекции, иллюстрирующий "висячую" ссылку. Выполните эту программу, переведите и выпишите сообщение об ошибке, которое будет получено. Выполните то же упражнение для процедур GetMem и FreeMem.
5. Напечатайте объём свободного пространства в куче при правильном захвате и освобождении динамических переменных, определив разницу между функциями MaxAvail и MemAvail.
6. Определите размер в байтах ссылочных типов в зависимости от типа, на который они указывают, множественных типов в зависимости от типа их элементов и различных файловых типов.
7. Выполните вычисление произведения и транспонирования матриц и алгоритм Уоршала построения транзитивного замыкания матрицы смежности произвольного графа для динамических массивов.

Методические рекомендации к выполнению:

Сравните возможности функции Addr и операции @.

Сопровождайте примеры иллюстрациями: переменные можно обозначать прямоугольниками, а ссылки на них – стрелками. Ссылку **nil** можно обозначать точкой внутри соответствующего прямоугольника.

Для однозначности при доступе к элементам структурных переменных воспринимайте запись разыменованного указателя A^ как единое имя переменной.

При выполнении 3-го задания необходимо учесть длину в байтах элемента массива каждого типа и свойство массивов Турбо Паскаля занимать сплошной отрезок памяти.

Чтобы увидеть разницу между функциями MaxAvail и MemAvail захватите, к примеру, две небольших динамических переменных, затем освободите первую из них.

Далее захватите более длинную переменную и только тогда освободите вторую небольшую переменную.
Обратите внимание на то, что аргумент функции SizeOf является **именем** только переменной или типа. В последнем случае функцию SizeOf можно использовать и в разделах описаний.
Для задания динамической матрицы её можно предварительно линеаризовать, выложив строки в динамический вектор. Тогда для доступа к элементу матрицы существенной величиной помимо индекса [I, J] будет размер (число элементов) её строки – N.

```
type      {Тип вектора, который будет хранить матрицу}
  TVect = array[1..65535 div SizeOf(TE)] of TE;
  PVect = ^TVect;
```

После захвата памяти под вектор Vect: PVect; доступ к элементу матрицы с индексом [I, J] ($1 \le I \le M$, $1 \le J \le N$) в реальности будет доступом к элементу вектора Vect^[N*(I - 1) + J].
Другой путь задания динамической матрицы заключается в использовании сочетания двух типов: TRow – тип строки матрицы, TCol – тип вектора-столбца указателей на строки.

```
type
  TRow = array[1.. 65535 div SizeOf(TE)] of TE;
  PRow = ^TRow;
  TCol = array[1.. 65535 div SizeOf(PRow)] of PRow;
```

Сами типы ориентированы на максимально возможные размеры индекса, поэтому память для динамической матрицы Matr: ^TCol; может захватываться без опасения выхода индекса за допустимые границы. Сначала захватывается динамический вектор-столбец указателей на строки (Matr^), а затем в цикле для каждого элемента столбца (Matr^[I]) – сами динамические строки (Matr^[I]^). Для доступа к элементам матрицы используется конструкция Matr^[I]^[J].