

Podstawy programowania komputerów – kolokwium

```
{dane pomocnicze}

const
  N = 6;

type
  PNode = ^TNode;
  TNode = record
    data : integer;
    next : PNode;
  end;

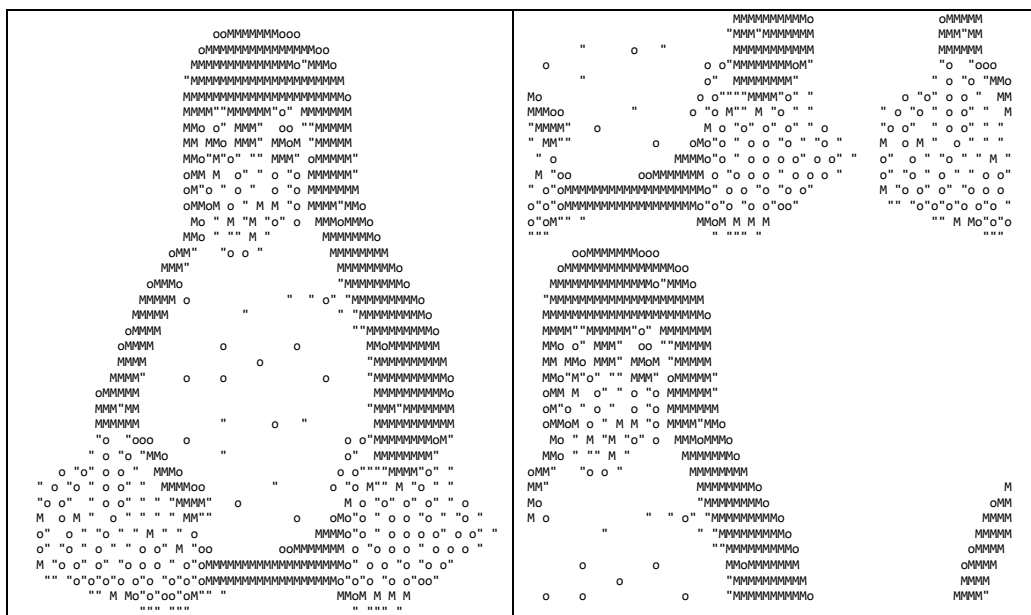
TPicture = array[0..N-1, 0..N-1] of char;
```

Zadanie 1 (4 p.)

Proszę napisać procedurę o następującym nagłówku:

```
procedure move(  const input : TPicture;
                  var output : TPicture;
                  x : integer;
                  y : integer);
```

Procedura ta dla zadanego obrazka typu ASCII-Art przekazanego przez argument `input` dokona jego przesunięcia o `x` pól w poziomie oraz `y` pól w pionie oraz zapisze wynik w tablicy `output`. Argumenty `x` oraz `y` mogą przyjmować dowolne wartości, także ujemne lub większe od `N`. Przesunięcie każdego symbolu poza obszar obrazka powinno spowodować jego umieszczenie po przeciwnej stronie. Przykład przesunięcia o -20 pól w poziomie oraz 15 w pionie:



Zadanie 2 (8 p.)

Dla zdefiniowanego typu węzłów listy jednokierunkowych `TNode` proszę napisać procedury o następujących nagłówkach:

```
procedure readFile(    const fileName : string;
                      var head : PNode);

procedure writeFile(  const fileName : string;
                      head : PNode);

procedure intersection(iHead1 : PNode;
                     iHead2 : PNode;
                     var oHead : PNode);

procedure clear(  var head : PNode);
```

Zadaniem wymienionych procedur jest kolejno:

- wczytanie listy z pliku rekordowego zawierającego wartości typu `integer`,
- zapis elementów listy do pliku rekordowego,
- znalezienie iloczynu zbiorów reprezentowanych przez listy `iHead1` oraz `iHead2` i zapisanie wyniku w liście `oHead`,
- usunięcie całej listy.

Można postawić następujące założenia:

- wykorzystanie plików nie spowoduje wystąpienia błędów,
- pliki wejściowe nie są puste, plik wyjściowy może zostać wyczyszczony,
- listy wyjściowe procedur `readFile` oraz `intersection` z początku są puste.