

Zad. 5 Dane są definicje:

```
punkt = rekord x,y : real; end;  
tab = array[1..max] of punkt;
```

Tablica typu tab zawiera współrzędne max punktów leżących na płaszczyźnie. Punkty posiadają jednostkową masę. Proszę napisać funkcję, która sprawdza czy istnieje niepusty podzbiór n punktów, gdzie $n < k$ oraz n jest wielokrotnością liczby 3, którego środek ciężkości leży w odległości mniejszej niż r od początku układu współrzędnych. Do funkcji należy przekazać dokładnie 3 parametry: tablicę typu tab, promień r, oraz ograniczenie k, funkcja powinna zwrócić wartość typu boolean.

Uwagi:

- oceniane będą: czytelność (komentarze), poprawność oraz efektywność programu;
- za zadanie można otrzymać max 2.5 pkt;
- czas na rozwiązanie 15 min.

```
function istnieje(var t:tab; r:real; k:integer):boolean;  
  
    function szukaj(p,n:integer; sx,sy:real):boolean;  
    begin  
        if (p>max) or (n>=k) then szukaj:=false  
        else if (n>1) and (n mod 3=0) and (sqrt(sqr(sx/n)+sqr(sy/n))<r) then szukaj:=true  
        else szukaj:=szukaj(p+1,n+1,sx+t[p].x,sy+t[p].y) or szukaj(p+1,n,sx,sy)  
        end  
    end  
  
begin  
    istnieje:=szukaj(1,0,0.0,0.0)  
end;
```

Zad. 6 Dany jest łańcuch zawierający liczby naturalne, zbudowany w oparciu o elementy typu:

```
pnode = ^node;  
node = record  
    klucz : integer;  
    next : pnode;  
end;
```

Proszę napisać procedurę, która rozdzieli elementy łańcucha wejściowego do 2 łańcuchów, zależnie od reszty dzielenia pola klucz przez 3. Dla reszty równej 1 lub 2, element należy umieścić odpowiednio w łańcuchu pierwszym lub drugim. Pozostałe elementy łańcucha wejściowego należy usunąć z pamięci. Do procedury należy przekazać wskazanie na łańcuch wejściowy, oraz wskazania na powstałe łańcuchy.

Uwagi:

- łańcuch wejściowy może być pusty;
- kolejność elementów w łańcuchach wyjściowych nie ma znaczenia;
- oceniane będą: czytelność (komentarze), poprawność oraz efektywność programu;
- za zadanie można otrzymać max 2.5 pkt;
- czas na rozwiązanie 15 min.

```
procedure rozdziel(var we,wyl,wy2 : pnode);  
var  
    p : pnode;  
begin  
    wyl:=nil;  
    wy2:=nil;  
    while we<>nil do begin  
        p:=we^.next;  
        case we^.klucz mod 3 of  
            0 : dispose(we);  
            1 : begin we^.next:=wyl; wyl:=we; end;  
            2 : begin we^.next:=wy2; wy2:=we; end;  
        end;  
        we:=p;  
    end  
end
```