

1. Proszę napisać w języku Pascal program wypisujący wartość ułamka A/B z dokładnością do N ($N > 1$) cyfr po kropce dziesiętnej. Wartości A, B i N program wczytuje z klawiatury. Dodatkowo, przy założeniu, że ułamek jest nieskracalny oraz mianownik nie zawiera w swoim rozkładzie 2 ani 5 wypisać długość okresu ułamka.

Rozwiązanie 1 (bez wyznaczanie długości okresu)

```
var
  a,b,n,i : integer;
begin
  read(a,b,n);           { wczytaj dane }

  write(a div b, '.');    { wypisz część całkowitą }
  a:=a mod b;             { przygotuj resztę }

  for i:=1 to n do begin  { wykonuj dzielenie „pisemne” }
    write(10*a div b);
    a:=10*a mod b;
  end;
end.
```

Rozwiązanie 2 (z wyznaczaniem długości okresu)

```
var
  a,b,n,i,okres,mem : integer;
begin
  read(a,b,n);           { wczytaj dane }

  write(a div b, '.');    { wypisz część całkowitą }
  a:=a mod b;

  mem:=a;                 { zapamiętaj resztę rozpoczynającą okres }
  i:=1;                   { licznik cyfr oraz licznik okresu }
  okres:=0;               { dopóki nie wyznaczysz długości okresu=0 }

  { pętla while zakończy się gdy wypiszemy odpowiednią liczbę cyfr
    oraz wyznaczymy długość okresu }

  while (i<=n) or (okres=0) do begin { wykonuj dzielenie pisemne }
    if i<=n then write(10*a div b); { wypisz kolejną cyfrę }
    a:=10*a mod b;
    if (a=mem) and (okres=0) then okres:=i; { powtórzenie reszty-koniec okresu }
    i:=i+1;
  end;
  writeln;
  writeln('okres=',okres);
end.
```

2. Proszę napisać w języku Pascal program wyliczający wartość logarytmu przy podstawie 2 z liczby naturalnej P z dokładnością $1/(2^N)$, gdzie N może wynosić 0,1,2,... Wartości P i N program wczytuje z klawiatury. W programie nie wolno korzystać z wbudowanych funkcji obliczających logarytm i potęgę.

Rozwiązanie obliczające logarytm z dokładnością do 1.0

```
var
  p,n : integer;
  l,w : real;

begin
  read(p,n);           { wczytaj dane }
  w:=1.0;              { tu obliczamy potęgę }
  l:=0.0;              { tu będzie wynik }
  while 2.0*w<=p do begin
    w:=w*2.0;
    l:=l+1;
  end;
  writeln(l, ' ',ln(p)/ln(2.0)); { wypisz wynik oraz wartość kontrolną }
end.
```

Rozwiązanie pełne.

```
var
  p,n,i : integer;
  l,w,m,s : real;

begin
  read(p,n);           { wczytaj dane }
  m:=2.0;              { na początek liczymy potęgi 2 }
  s:=1.0;              { to dodajemy do wyniku }
  w:=1.0;              { tu obliczamy potęgę }
  l:=0.0;              { tu będzie wynik }
  for i:=0 to n do begin
    while w*m<=p do begin w:=w*m; l:=l+s; end;
    m:=sqrt(m);
    s:=s/2.0;          { „schodzimy” z dokładnością o połowę }
  end;
  writeln(l, ' ',ln(p)/ln(2.0)); { wypisz wynik oraz wartość kontrolną }
end.
```

Proszę napisać w języku Pascal program obliczający i wypisujący wartość podstawy logarytmu naturalnego z dokładnością do n cyfr po przecinku (n jest rzędu 100). Wartość n powinna być parametrem programu. W obliczeniach należy skorzystać ze wzoru $e = 1/0! + 1/1! + 1/2! + 1/3! + 1/4! + \dots$

```
const
    n=100;
var
    i,j,p,r : integer;
    koniec : Boolean;
    s,sum : array[0..n] of integer;

begin
    { s := 0, sum:=0 }
    for i:=0 to n do begin
        s[i]:=0;
        sum[i]:=0;
    end;
    s[0]:=1;      { s := 1 }
    sum[0]:=1;    { sum := 1 }

    i:=1;
    repeat
        koniec:=true;
        { s := s/i }
        r:=0;
        for j:=0 to n do begin
            r:=(10*r+s[j]);
            s[j]:=r div i;
            if s[j]>0 then koniec:=false;
            r:=r mod i;
        end;

        { sum := sum+s }
        p:=0;
        for j:=n downto 0 do begin
            p:=sum[j]+s[j]+p;
            sum[j]:=p mod 10;
            p:=p div 10;
        end;

        inc(i);
    until koniec;

    { wypisz wynik }
    write(sum[0],'.');
    for i:=1 to n do write(sum[i]);
    writeln;
end.
```

1. Dany jest typ tablicowy **mapa = array [1..max, 1..max] of Boolean**; reprezentujący mapę. Wartość true oznacza ląd, a wartość false ocean. Na oceanie znajdują się wyspy utworzone przez sąsiadujące pola (rysunek). Proszę napisać w Pascalu funkcję do której przekazujemy mapę i współrzędne x,y punktu na mapie. Jeżeli współrzędne oznaczają punkt na wyspie, funkcja powinna zatopić wyspę. Funkcja powinna zwrócić rozmiar zatopionego obszaru. Można założyć, że żadna wyspa na mapie nie styka się z jej krawędzią.

```
Function woda(var m:mapa; x,y:integer):integer;
begin
  woda:=0;
  if m[x,y] then begin
    m[x,y]:=false;
    woda:=1+woda(m,x+1,y)+woda(m,x-1,y)+woda(m,x,y+1)+woda(m,x,y-1);
  end;
end;
```

2. Dany jest łańcuch odsyłaczowy o niepowtarzalnych kluczach:

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

Napisać procedurę do której przekazujemy wskaźnik na początek łańcucha oraz wartość klucza. Jeżeli element o takim kluczu występuje w łańcuchu należy go usunąć z łańcucha. Jeżeli elementu o zadanym kluczu brak w łańcuchu należy element o takim kluczu wstawić do łańcucha.

```
procedure dodaj_lub_usun(var p:pnode; klucz:integer);
var
  q,r:pnode;
begin
  q:=p;
  r:=nil; { wskaźnik na poprzedni element }
  while (q<>nil) and (q^.klucz<>klucz) do begin
    r:=q; q:=q^.next;
  end;
  if q=nil then begin { wstawianie elementu na początku łańcucha }
    new(q); q^.klucz:=klucz; q^.next:=p; p:=q;
  end else begin { usuwanie elementu wskazywanego przez q }
    if q=p then p:=p^.next { jeżeli jest to pierwszy element łańcucha }
    else r^.next:=q^.next; { jeżeli dalszy element łańcucha }
    dispose(q);
  end;
end;
```

I kartkówka (obie grupy)

- napisz program, który wypisuje trójki pitagorejskie, mniejsze od n

II kartkówka - TABLICE (grupa a)

- dana jest 2-wymiarowa tablica integer, znajdź element dla którego suma otaczających go elementów będzie największa

II kartkówka - TABLICE (grupa b)

- dana jest 2-wymiarowa tablica integer, znajdź element dla którego suma elementów leżących na przekątnych na których przecięciu znajduje się nasz element jest największa

III kartkówka - REKORDY (obie grupy)

- dana jest 1-wymiarowa tablica rekordów, w których zapisane są współrzędne x, y punktów, napisz funkcję zwracającą liczbę punktów posiadających w otoczeniu r więcej niż 77 sąsiadów

IV kartkówka - REKURENCJA (obie grupy)

- dana jest 2-wymiarowa kwadratowa tablica integer, napisz funkcję rekurencyjną, która wypisze najkrótszą drogę i jej długość, przy przechodzeniu z punktu $(1,1)$ do (n,n) i poruszaniu się jedynie w dół i prawo

V kartkówka - WSKAŹNIKI (grupa a)

- dana jest uporządkowana lista 2-kierunkowa, napisz procedury lub funkcje: dodawania elementu do listy za jakiś element i wstawiania elementu do listy na pierwsze miejsce, na ich podstawie napisz procedurę lub funkcję, która doda element w odpowiednie miejsce

V kartkówka - WSKAŹNIKI (grupa b)

- dana jest uporządkowana lista 2-kierunkowa, napisz procedury lub funkcje: dodawania elementu do listy przed jakiś element i wstawiania elementu do listy na ostatnie miejsce, na ich podstawie napisz procedurę lub funkcję, która doda element w odpowiednie miejsce

VI kartkówka - WSKAŹNIKI + REKURENCJA (obie grupy)

- napisz funkcję rekurencyjną, która utworzy 1-kierunkową listę o kolejnych elementach naturalnych mniejszych bądź równych n