Priznávam, že s tým kvízom o Pascale som to na prvý šup poriadne prestrelil. Dovoľte mi preto, aby som aspoň takto skúsil vysvetliť, o čo mi v jednotlivých otázkach išlo.

Aj keď výsledková listina tohoto kvízu vyzerá skutočne dramaticky, takto dodatočne si uvedomujem, že väčšina - ak nie všetci - účastníci kvízu *nie* sú praktizujúcimi programátormi v tomto jazyku. No a väčšina otázok bola skutočne veľmi ťažká, zabiehajúc aj do menej používaných detailov jazyka, a boli medzi nimi aj nefalšované chytáky.

Takže poďme pekne po poriadku:

1. Autor jazyka Pascal

Autorom jazyka Pascal je:

a) Blaise Pascal
b) Niclaus Wirth
c) Dennis Ritchie
d) Thomas Edison
e) Richard Stallman

Toto bola zahrievacia otázka, a myslel som si, že k odpovedi sa dá dospieť aj vylučovacím spôsobom: *Blaise Pascal* bol francúzsky fyzik, ktorému sa pripisuje (okrem iných serióznejších vecí) vynález rulety a pozičného mechanického počítacieho stroja s prenosom medzi rádmi (t.j. to čo máme dnes v každom elektro- plyno- a iných -meroch); po ňom je jazyk pomenovaný. *Dennis Ritchie* je hlavným autorom jazyka C a spoluautorom známej referenčnej príručky o tomto jazyku. [i]Thomas Edison[i] je menovec webmastra uZone

2. Kľúčové slová

Kľúčové slová sú vyhradené reťazce, ktoré majú kľúčovú úlohu v definícii jazyka, a teda nie je možné ich predefinovať a použiť pre potreby aplikácie napr. ako názov premennej.

Ktoré z nasledujúcich slov sú kľúčovými slovami v Pascale (viacnásobná odpoveď)

a) const
b) typedef
c) set
d) reset

Kľúčové slovo *const* uvádza sekciu, kde sa definujú v programe konštanty, napr.

1. const
```
pi = 3.123456;
```
2.
```
meno = 'Jano';
```
3.

typedef je kľúčové slovo z C (t.j. pokus o chyták), a v Pascale nemá žiadnu vyhradenú úlohu (t.j. "typedef" si môže užívateľ použiť ako chce).

Kľúčové slovo *set* (v spojení *set of*) je určené na tvorenie typu množina. Tento silne abstraktný typ je napríklad programátorom v jazyku C úplne neznámy, a slúži napríklad na výber nejakej podskupiny vlastností z vopred definovaného zoznamu, a následne na operácie zisťujúce, či niektorá vlastnosť v danej podskupine je alebo nie je.

Slovo *reset* nebude užívateľom Pascalu neznámy, a vzhľadom na jeho časté použitie by sa mohlo zdať, že sa jedná o kľúčové slovo. Je názov funkcie systémovej knižnice, ktorá slúži na otvorenie súboru na čítanie. Nie je to však kľúčové slovo, je možné toto slovo použiť napr. ako názov premennej, alebo definovať vlastnú funkciu s týmto menom, hoci z pochopiteľných dôvodov sa ani jedno z toho nedoporučuje.

3. Zápis čísla

Hexadecimálne číslo zodpovedajúce decimálnemu číslu 28 sa zapisuje:

a) $1C
b) $1c
c) 0X1C
d) 0x1c
e) 01Ch

Hexadecimálne konštanty ("literals") sa v Pascale zapisujú s prefixom *$*. Pascal je "case insensitive" (nezáleží v ňom na veľkosti znakov), preto je aj $1C aj $1c správne.

Prefix *0x* je známy napríklad z jazyka C, ktorý je však "case sensitive". (Na túto tému mi nedá nespomenúť klasický chyták jazyka C, a to že samotná 0 na začiatku konštanty je pokladaná za prefix osmičkovej sústavy).

Suffix *h* je na označenie hexadecimálnych konštánt používaný napr. v asembleroch Intelovských mikrokontrolérov.

4. Procedúry

Majme program:

```
1.
   program test;
2.
   var p: integer; q: integer;
3.
4.
   procedure p1(var a: integer; b:                integer);
5.
   begin
6.
     b := 2 * b;
7.
     a := 3 * a;
8.
   end;
9.
10.
   begin
11.
     p := 2;
12.
     q := 3;
13.
     p1(p, q);
14.
     writeln('p = ',p,              ' q = ',q);
15.
   end.
```

16.

Čo sa vypíše?

a) p = 6 q = 3
b) p = 2 q = 3
c) p = 6 q = 6
d) p = 4 q = 9

V tomto príklade išlo o dva rôzne spôsoby odovzdávania parametru procedúre: odovzdávanie referenciou (keď je v hlavičke funkcie pred formálnou deklaráciou parametra kľúčové slovo *var*, v našom prípade je to parameter *a*), a odovzdanie hodnotou (bez *var*, v našom príklade parameter *b*).

Pri odovzdaní *referenciou* sa akákoľvek manipulácia s parametrom vykonáva priamo na premennej, ktorá bola do tohoto parametra priradená pri volaní. Nie je teda samozrejme možné použiť pri volaní na takomto mieste konštantu či výraz. Programátori v jazyku C takéto volanie nahrádzajú odovzdávaním ukazovateľa, s patričnými negatívnymi následkami na prehľadnosť apod.

Pri odovzdaní *hodnotou* sa jednoducho vytvorí nová lokálna premenná, do ktorej sa priradí odovzdávaná hodnota - toto je spôsob odovzdania ktorý programátori v jazyku C poznajú. Zmeny v tejto premennej sa nijako neprenášajú na premennú vo volaní, a po ukončení procedúry automaticky zaniká.
V našom prípade sa teda odovzdala premenná p do parametra a referenciou, a pri spätnom uložení vynásobenej hodnoty parametra a sa v skutočnosti zmení hodnota premennej p, z pôvodných 2 na trojnásobok, t.j .6. Premenná q sa odovzdala do parametra b hodnotou, takže na ňu operácie v procedúre nemajú žiadny vplyv, jej hodnota zostala nezmenená, t.j. 3.

Pokus o chyták spočíval jednak v opačnom poradí operácií v procedúre než sú parametre deklarované (to je ozaj úbohý pokus 
), a tiež v opticky rovnakej deklarácii parametrov než sú deklarované premenné p a q (čo sa pokúšalo nanútiť predstavu, že sa aj s p aj s q nakladá rovnako).

5. Booleovský výraz

V príklade:

```
1.
    var a, b, c, d: boolean;
2.
3.
    begin
4.
      a := false;
5.
      b := false;
6.
```

```
        c := false;
7.
        d := false;
8.
        if a = b or c and d then
9.
          {vetva po then}
10.
        else
11.
          {vetva po else}
12.
        ;
13.
      end.
14.
```

sa vykoná vetva:

a) vetva po then
b) vetva po else
c) skončí runtime error-om
d) nepreloží sa kvôli syntaktickej chybe

Precedencia operátorov v Pascale je pomerne neintuitívna, napríklad *and* a *or* majú rozdielnu precedenciu. Najvyššiu prioritu má and, potom or, a až potom porovnanie, takže sa tento výraz vyhodnocuje pomerne neintuitívne sprava: false and false dá false, toto or false dá false, a toto porovnané s false dá true.

Častým súvisiacim problémom je napríklad zápis typu
if a > 3 and b < 4
kde je vďaka najvyššej priorite vyhodnotený operátor and, čím však vznikne nevyhodnotiteľný výraz s dvomi porovnaniami.

Typ *boolean* je natívnym typom v Pascale od prvej verzie jazyka; na rozdiel od C, kde je podpora pre booleovské typy pridaná do štandardu len pomerne nedávno, a aj to zaťažená neefektivitou a rôznymi komplikáciami najmä z historických a kompatibilných dôvodov.

6. Cykly

V nasledujúcom príklade:

```
1.
2.
```

```
var k:integer;
3.
begin
4.
 k := 5;
5.
 repeat
6.
  {telo cyklu}
7.
  k := k - 1
8.
 until k > 0;
9.
end.
10.
```

vykoná

a) ani raz
b) raz
c) štyrikrát
d) päťkrát
e) program sa nepreloží kvôli syntaktickej chybe

Tu mi z textu vypadlo *sa telo cyklu [vykoná]*, za čo sa dodatočne ospravedlňujem - snáď bola úloha aj takto dostatočne zrozumiteľná.

Cyklus repeat-until sa vždy vykoná aspoň raz, keďže sa podmienka vyhodnocuje na konci; a opustí sa, ak má podmienka výsledok true. Toto je v tomto prípade splnené hneď pri prvom prechode, keďže na konci prvého prechodu telom cyklu je k = 5 - 1 = 4 a to je > 0.

Chyták tu spočíval v navodení dojmu, že sa cyklus vykoná 5 krát (čo by s touto podmienkou platilo v C-čkovom cykle do-while).

7. Bonusová otázka

Program v Pascale je ukončený

a) bodkou
b) prázdnym riadkom
c) tromi prázdnymi riadkami
d) end;

Myslel som si, že toto je najľahšia otázka, preto som ju nazval bonusovou...

wek

To *end.* (t.j. end s bodkou, na rozdiel od end s bodkočiarkou na konci každého iného bloku) na konci programu je totiž úplne enigmatická záležitosť - dokonale zbyttočný štylistický detail, ktorý však je až s maniakálnou dôslednosťou kontrolovaná bežnými prekladačmi.

===
Dúfam, že aj napriek všetkému ste nezanevreli na uZone, kvízy a Pascal. Pokúsim sa to napraviť v ďalšom Pascalovskom kvíze, a sľubujem, že tentoraz bude ľahší...