Četrte vaje iz računalništva I

Jurij Rici

3. 11. 1998

Od zadnjič: Naredili smo program, ki je odčital število, ki je omejeno - je tipa int. To število bi radi zapisali, tako da se izpiše vsaka števka števila z besedami in sicer v obratnem vrstnem redu. Npr. če imamo število 173, bi radi dobili `tri sedem ena`.

Vprašamo se:

- Kako bi to sedaj naredili?
- Kaj in kje moramo to spremeniti?

Namig: Uporabi `if` stavek.

Kakšna je oblika `if` stavka ?

Ta stavek ima sledečo obliko. Zapišemo:

```
if (pogoj)
{
  stavek ;
};
```

Če je pogoj zadovoljen (izpolnjen), se bo `stavek` izvedel.

Izpis števk z besedami

Definirali bomo funkcijo, ki bo števke izpisovala z besedami.

Funkcije – še enkrat

Vsaka funkcija ima *telo* in *glavo*: v glavi funkcije opišemo *kaj* funkcija počne in v in v njenem opišemo *kako* to počne.

Poleg tega ima funkcija tudi dve *pojavitvi*: *definicijo* in *uporabo* (slednja je seveda lahko večkratna). Drugače povedano v prvi pojavitvi funkcije funkcijo opišemo, v drugi pa jo uporabimo – jo kličemo.

Naša funkcija

Zanima nas, ali bo funkcija kaj vrnila? Ne in zato je tipa `void`.

```
void izpis_stevke (int stevka)
{
  if (stevka == 0)  prinf(" nic");
```

```
  if (stevka == 1)  prinf(" ena");
  if (stevka == 2)  prinf(" dve");
  if (stevka == 3)  prinf(" tri");
  if (stevka == 4)  prinf(" stiri");
  if (stevka == 5)  prinf(" pet");
  if (stevka == 6)  prinf(" sest");
  if (stevka == 7)  prinf(" sedem");
  if (stevka == 8)  prinf(" osem");
  if (stevka == 9)  prinf(" devet");
} /* izpis_stevke */
```

Če želimo, da nam program začne novi zagon v novi vrsti, moramo nekoliko popraviti funkcijo `izpisi`.

Zakaj funkcijo `izpisi`? Zato ker nam ona izpisuje števke.

Kje moramo popraviti? Funkcija `izpisi` najprej naracuna količnik in stevko. Nato po vrsti izpise stevko in se preostali del stevila (količnik). V primeru, ko je količnik enak 0, ni potrebno izpisati ničesar več, ampak samo še zaključimo izpisovanje s prehodom v novo vrstico:

```
if (kolicnik != 0)
  izpisi(kolicnik);
else
  printf("\n");
```

Izpisovanje števke malo drugače

Sedaj bi radi naredili to funkcijo nekoliko drugače. Uporabili bom `switch` stavek.

Kakšno obliko ima switch stavek? Ta stavek ima sledečo obliko (glej tudi pomoč (*HELP*) v programu):

```
switch (spremenljivka)
{
case vrednost_1: stavek_1; break;
case vrednost_2: stavek_2; break;
...
case vrednost_n: stavek_n; break;
}
```

Če je vrednost spremenljivka `vrednost_i` (i=1,2,...,n), se izvede `stavek_i`.

Kaj pa če imamo dve vrednosti hkrati? Potem se izvedeta obe.

Izvedejo se vsi stavki od prvega navzdol. Če mislimo, da je to nepotrebno, uporabimo `break` stavek. Ko pridemo do ustreznega stavka, nam ukaz `break` prekine izvajanje `switch` stavka. Torej izvede se tisti del programa, ki ustreza stavkom. Tako dobimo:

```
void izpisi_stevko(int stevka)
{
  switch (stevka)
  {
  case 0: printf(" nic"); break;
  case 1: printf(" ena"); break;
```

```
  case 2: printf(" dve"); break;
  case 3: printf(" tri"); break;
  case 4: printf(" stiri"); break;
  case 5: printf(" pet"); break;
  case 6: printf(" sest"); break;
  case 7: printf(" sedem"); break;
  case 8: printf(" osem"); break;
  case 9: printf(" devet");
  }/* switch */
} /* izpisi_stevko */
```

Izpisovanje števil še drugače

Kaj pa če želimo števke zapisati v istem vrstnem redu, kot so bile vpisane (to je `ena sedem tri`)? Kateri del programa moramo spremeniti?

Spremeniti moramo funkcijo `izpisi`. Funkcija `izpisi` nam izpisuje števke. Kako to počne?

Ker nam funkcija najprej izpisuje števko, potem pa količnik (preostanek števila), moramo zamenjati vrstni red: najprej mora izpisuje količnik in šele nato števko.

Kaj pa če želimo števke zapisati v istem *in* obratnem vrstnem redu? V tem primeru izpišemo števko, nato izpišemo količnik (za izpis tega bo poskrbela funkcija `izpisi`) in na koncu še enkrat izpišemo števko.

Na koncu pa želimo popraviti še ves program tako, da se bo izvedel večkrat zaporedoma, ne da bi ga zaganjali vsakič posebej. Pri tem pa bi se izvajanje prekinilo šele takrat, ko bi odtipkali določeno število. Uporabimo zanko `while` v glavnem programu.