

Zapiski s priprav
Računske operacije

ACM priprave na tekmovanja

Različica z dne 23. oktober 2024

Kazalo

1	Seštevanje, odštevanje in množenje	2
1.1	Negativna števila	2
2	Deljenje	3

1 Seštevanje, odštevanje in množenje

Najpreprostejše računske operacije na številih, ki jih lahko z računalnikom izračunamo, so seštevanje, odštevanje in množenje. Račune zapisujemo tako kot v šoli, z *operatorji*. Za seštevanje uporabimo +, za odštevanje - in za množenje *. Poglejmo si preprost primer, kjer račun izvedemo kar v funkciji printf.

```
1  #include <stdio.h>
2
3  int main(){
4      int a = 5, b = 7;
5      printf("%d\n", a+b);
6      return 0;
7  }
```

Zgornji program preprosto izpiše številko 12, ki je rezultat računa 5 + 7. Namesto izpisovanja lahko rezultat tudi shranimo v spremenljivko:

```
1  #include <stdio.h>
2
3  int main(){
4      int a, b, vsota, razlika, produkt;
5      scanf("%d%d", &a, &b);
6      vsota = a+b;
7      razlika = a-b;
8      produkt = a*b;
9      printf("%d\n%d\n%d\n", vsota, razlika, produkt);
10     return 0;
11 }
```

Če v ta program podamo vhod 3 7, bo izpisal tri števila: 12, -4 in 21. Drugo izpisano število ima pred seboj minus. Če te to preseneča in še ne poznaš negativnih števil, si preberi naslednji razdelek, sicer pa ga lahko izpustiš.

1.1 Negativna števila

Na meteorološki postaji Kredarica so leta 2014 izmerili povprečno januarsko temperaturo približno -5°C , povprečno avgustovsko pa približno 6°C . Med tema meritvama je 11°C razlike. Pozimi lahko izmerimo temperature manjše od 0. Takšnim številom, kot je -5 , rečemo *negativna števila*. Lahko pa jih uporabimo tudi drugje, ne samo pri merjenju temperature. S pozitivnimi števili lahko štejemo od 0 do neskončno (1, 2, 3, ...),

z negativnimi pa do negativne neskončnosti $(-1, -2, -3, \dots)$. Tako kot pozitivna števila jih lahko seštevamo in odštevamo:

$$\begin{aligned}5 - 11 &= -6 \\ -6 + 11 &= 5 \\ 5 - (-6) &= 11 \\ -2 - 1 &= -3 \\ -2 - (-1) &= -1 \\ -2 + (-1) &= -3\end{aligned}$$

Pri tem se sklicujemo na naslednji pravili, kjer smo z x označili poljubno število:

$$\begin{aligned}-(-x) &= x \\ +(-x) &= -x\end{aligned}$$

Z negativnimi števili lahko tudi množimo:

$$\begin{aligned}2 \cdot (-5) &= -10 \\ (-5) \cdot (-5) &= 25\end{aligned}$$

Pri tem uporabimo enostavno pravilo: številski del rezultata je enak, kot če bi množili števili brez predznaka $-$. Če smo množili eno pozitivno in negativno število, rezultatu pripišemo še negativni predznak, če pa smo množili dve negativni števili ali dve pozitivni števili, pa tega ne storimo. Računalnik pri računanju ne razlikuje med pozitivnimi in negativnimi števili, računske operacije pišemo enako kot pri pozitivnih.

2 Deljenje

Števila lahko tudi delimo, za kar uporabimo znak $/$. Pri tem pa moramo biti pozorni, ker se deljenje v računalniku obnaša drugače kot smo navajeni iz matematike. Običajno pri deljenju dveh števil, recimo 3 in 7, dobimo ulomek:

$$3/7 = \frac{3}{7}.$$

Če pa v C++ program zapišemo `printf("%d\n", 3/7)`, bomo dobili nepričakovan odgovor `— 0`. To je zato, ker je deljenje v C++ *celoštevilsko*.

Za lažje razumevanje si bomo pomagali s formulo $a = k \cdot b + o$, ki ponazarja deljenje z ostankom. Če želimo število a deliti s številom b , bomo kot odgovor dobili dve števili: celi del, ki smo ga uspešno delili, in ostanek, kjer deljenje ni bilo uspešno. Celi del smo zgoraj označili s k , ostanek pa z o . Ko uporabimo operator $/$, dobimo največje tako število, za katero je produkt rezultata in delitelja (števila na desni strani $/$) manjši ali

enak deljencu (številu na levi strani /). Z drugimi besedami, dobimo k iz zgornjega zapisa. Če želimo preveriti še vrednost ostanka o , jo lahko izračunamo iz ostalih števil,

$$o = a - k \cdot b,$$

ali pa uporabimo poseben operator %, ki mu pravimo *modulo*. Poglejmo si, kako deljenje deluje v praksi.

```
1  #include <stdio.h>
2
3  int main(){
4      int a, b;
5      scanf("%d%d", &a, &b);
6      int k = a / b, o = a % b;
7      printf("rezultat deljenja je %d, z ostankom %d\n", k, o);
8      printf("%d = %d * %d + %d\n", a, k, b, o);
9      return 0;
10 }
```

Če v zgornji program vpišemo dve števili, bo izpisal rezultat deljenja ter razcep števila po zgornji formuli. Na primer, ob vходу 25 7 dobimo naslednji izpis:

```
rezultat deljenja je 3, z ostankom 4
25 = 3 * 7 + 4
```

če pa vpišemo 3 7, dobimo

```
rezultat deljenja je 0, z ostankom 3
3 = 0 * 7 + 3
```

Kaj pa se zgodi, če vpišemo 5 0? V tem primeru bo program poskusil deliti z 0, kar v matematiki (in v programiranju) ni dovoljeno. Program se bo zato sesul in ne bo izpisal ničesar, odvisno od našega operacijskega sistema pa morda dobimo kakšno sporočilo o napaki. Ko delimo ali računamo ostanek z modulom, in je število na desni strani operatorja spremenljivka, katere vrednosti ne poznamo vnaprej, moramo preveriti, če je slučajno enaka 0. Če je, operacije ne smemo izvesti!