Branje in pisanje

1 Vhod in izhod

Programi za svoje delovanje potrebujejo način za komunikacijo z uporabnikom. Kompleksnejši programi v ta namen uporabljajo ekran, miško in tipkovnico. Pri tekmovalnem programiranju pa najpogosteje uporabljamo najpreprostejši način za komunikacijo: pisanje in branje s *standardnega vhoda in izhoda*. Običajno to pomeni, da se nam ob zagonu programa odpre okno, kamor lahko pišemo programu in kamor program izpisuje stvari.
Ko želimo, da naš program kaj izpiše, uporabimo *funkcijo* `printf`.

Primer

```
#include<stdio.h>

int main(){
        printf("Hello World!\n");
        return 0;
}
```

Primer vhoda in izhoda

```
Hello World!
```

`printf` - funkcija, ki ji v dvojnih narekovajih damo besedilo ali števila, ki jih želimo izpisati
`\n` - znak za novo vrstico

Stvari, ki jih mora vsebovati (skoraj) vsak program:

- `stdio.h` - *knjižnica* (datoteka), ki vsebuje funkcije, ki jih bomo uporabljali v programu (kot npr. `printf` in `scanf`)
- `#include<>` - ukaz, s katerim našemu programu povemo, katere knjižnice potrebuje
- `int main(){}` - telo našega programa - večino kode v programu napišemo med zavite oklepaje
- `return 0` - zadnja vrstica v programu, ki sporoča računalniku, da se je pravilno zaključil

V program lahko dodamo *komentarje*. To je takšno besedilo, ki je napisano v kodi, a vsebinsko ne vpliva na program.

- `//` - s tem zakomentiramo vse od poševnic do konca vrstice
- `/* */` - s tem lahko zakomentiramo več vrstic ali del znotraj vrstice

Primer

```c
#include<stdio.h>

int main(){ //komentar do konca vrstice
        printf /*komentar znotraj vrstice*/("Zivjo svet!\n");
        /*
        komentar
        čez
        več
        vrstic
        */
        return 0;
}
```

Primer vhoda in izhoda

```
Zivjo svet!
```

Pogoste napake

Zadnji znak, ki ga program izpiše, mora biti `\n`.

Pogoste napake

Večina vrstic v programu se konča s podpičjem (`;`) (skoraj vse razen tistih, ki se končajo z oklepaji ali zavitimi zaklepaji). Brez tega program ne bo delal.

2 Branje

Program za branje stvari, ki mu jih sporočamo, uporablja funkcijo `scanf`.

Primer

```c
#include<stdio.h>

int main(){
        char ime[50];
        printf("Kako ti je ime?\n");
        scanf("%s", ime);
        printf("Zivjo, %s!\n", ime);
        return 0;
}
```

Primer vhoda in izhoda

output: Kako ti je ime?
input: Tinka
output: Zivjo, Tinka!

Če želimo, da program lahko kaj počne s podatki, ki jih je prebral, moramo najprej to shraniti na neko mesto v spominu. Temu mestu rečemo *spremenljivka*, saj lahko s programom spreminjamo, kaj je tam shranjeno. Spremenljivke v svojem programu poimenujemo, v našem primeru ji rečemo `ime`. Vsaki spremenljivki moramo določiti *podatkovni tip*, saj lahko beremo in pišemo več različnih vrst podatkov, npr. besede ali števila.

`char` - s tem povemo, da je naša spremenljivka besedilo oz. *niz*

`[...]` - številka v oglatih oklepajih za besedo pove največjo dolžino niza, ki ga lahko program prebere.

Pogoste napake

Ko določamo največjo dolžino niza, vedno vzamemo večjo številko, kot jo bomo potrebovali. O tem bomo več govorili v kasnejših poglavjih.

Funkciji `scanf` podamo dva ali več *parametrov*:

- kaj naj prebere, torej kakšne tipe spremenljivk. To podamo s *formatnikom* (v našem primeru `%s`, ki pomeni niz (*string*). `%s` prebere vse znake do prvega presledka ali nove vrstice)
- ostali parametri povejo, kam naj funkcija shrani stvari, ki jih je prebrala (torej v spremenljivke, ki smo jih naredili prej)

Do zdaj smo funkciji `printf` podali samo točno določeno besedilo, ki smo ga želeli izpisati. Izpisujemo pa lahko tudi spremenljivke, kot smo to naredili v tem zadnjem primeru. Znotraj besedila dodamo formatnike na mesta, kjer želimo, da so spremenljivke, potem pa izven narekovajev naštejemo imena spremenljivk, ki jih želimo izpisati (tako kot pri funkciji `scanf`).

3 Branje števil

Poleg nizov lahko programi delajo tudi s števili. Števila beremo in izpisujemo z istima funkcijama kot nize, vendar moramo uporabiti drug formatnik.

Primer

```
#include<stdio.h>

int main(){
        int razred;
        printf("Kateri razred si?\n");
        scanf("%d", &razred);
        printf("%d. razred je najboljši.\n", razred);
        return 0;
}
```

Primer vhoda in izhoda

```
output:  Kateri razred si?
input :  7
output:  7.  razred je najboljši.
```

`&` - znak, ki ga moramo dati pred ime spremenljivke vedno, kadar beremo števila. `int` - podatkovni tip število
Pri številih za imenom spremenljivke ne povemo, kako velika so lahko.

Za branje in pisanje števil uporabimo formatnik `%d`.

Pogoste napake

Ko beremo števila, moramo pred ime spremenljivke dati znak `&`, česar pri branju nizov ne delamo. Prav tako tega ne delamo pri izpisovanju števil.