

INFORMATIČKI KLUB
FUTURA

```
#include<stdio.h>
int main()
{
RADIONICE PROGRAMIRANJA
    printf("Hello World!");
ZA SREDNJOŠKOLCE I STUDENTE
    return 0;
}
```

**RADIONICE PROGRAMIRANJA ZA SREDNJE
ŠKOLE 5. RADIONICA - POČETNICI**

Informatički klub FUTURA
Dubrovnik, 2017.



Creative Commons



□ slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo



□ pod slijedećim uvjetima:

- **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Pokazivač (engl. pointer)

- Pokazivač je varijabla koja sprema memorijsku lokaciju neke druge varijable (pokazuje na drugu varijablu).
- Pokazivaču se može mijenjati vrijednost, tj. može se mijenjati na koju memorijsku lokaciju pokazuje.

- Kao i varijabla, sastoji se od:
 - - tip - isti kao tip varijable na koju pokazuje. Dodaje se znak * da se označi da se radi o pokazivaču. Npr. int*
 - - ime
 - - vrijednost – memorijska lokacija varijable na koju pokazuje

Pokazivač (engl. pointer)

- //Primjer:
- #include <stdio.h>

- int main() {
- int broj1 = 5;
- int broj2 = 10;
- int* pokazivac = &broj1;//pokazivač gleda na memorijsku lokaciju varijable broj1. Znak & oznacuje memorijsku lokaciju/adresu varijable

- printf("%d %d %d\n", broj1, broj2, *pokazivac);//vrijednosti varijabli
- printf("%d %d %d\n", &broj1, &broj2, pokazivac);//memorijske lokacije

- *pokazivac = 7;

- printf("%d %d %d\n", broj1, broj2, *pokazivac);//vrijednosti varijabli
- printf("%d %d %d\n", &broj1, &broj2, pokazivac);//memorijske lokacije

- pokazivac = &broj2;

- printf("%d %d %d\n", broj1, broj2, *pokazivac);//vrijednosti varijabli
- printf("%d %d %d\n", &broj1, &broj2, pokazivac);//memorijske lokacije

- return 0;
- }

Pokazivač (engl. pointer)

- Koristi se:
- -kada je potrebno pristupiti varijabli izvan njezinog djelokruga (engl. scope).
- -za dinamičke varijable (varijable koje se koriste određeni dio vremena, pridruži im se neko područje u memoriji koje se oslobodi kada se završi rad s njima, npr. čitanje tekstualne datoteke)

□ Primjer:

□ `#include <stdio.h>`

□ `void postaviNa10(int* broj) { //primamo varijablu preko pokazivaca`

□ `*broj = 10; //mijenjamo vrijednost varijable x preko pokazivača broj izvan njezinog djelokruga`

□ `}`

□ `int main() {`

□ `int x = 1;`

□

□ `postaviNa10(&x); //saljemo adresu varijable x pokazivacu`

□

□ `printf("%d\n", x);`

□

□ `return 0;`

□ `}`

Zadatak 1

- ❑ Izračunajte zbroj i razliku dvije varijable s pomoću funkcija `zbrojDvaBroja` i `razlikaDvaBroja`.
- ❑ Funkcija `zbrojDvaBroja` mora primiti varijablu u koju će spremiti zbroj kao argument.
- ❑ Funkcija `razlikaDvaBroja` mora vratiti razliku dva broja.

Polje kao parametar funkcije

- ❑ Polje kao parametar funkcije možemo slati na više načina.
- ❑ Može se poslati kao polje:
 - ❑ `void funkcija(int polje[][10], int brojRedaka, int brojStupaca)//deklaracija`
 - ❑ `funkcija(polje, 10, 10);//pozivanje`
 - ❑ `polje[i][j]//korištenje`
- ❑ Može se poslati kao pokazivač na prvog člana:
 - ❑ `void funkcija(int *polje, int brojRedaka, int brojStupaca)//deklaracija`
 - ❑ `funkcija(&polje[0][0], 10, 10);//pozivanje`
 - ❑ `polje[i*brojStupaca+j]//koristimo kao 1d polje`
- ❑ Ako je polje poslano preko pokazivača, možemo pristupiti svakom članu pomoću `*polje`, ali tada je potrebno pomicati pokazivač, npr. `polje++`

Zadatak 2

- Generirajte polje slučajnih brojeva (u intervalu $[1, 100]$). Polje treba imati 3 redka i 5 stupaca. Ispišite polje i izračunajte koliko ima parnih, a koliko neparnih elemenata. Sve operacije na polju je potrebno raditi u funkcijama.

Zadatak 3

- Napišite funkciju koja ce zadani niz pretvoriti u mala odnosno velika slova.
- Funkcija treba biti oblika:
 - `void pretvori(char *niz, int malo);`
- gdje 0 znači pretvori u mala slova, a 1 pretvori u velika slova.

Zadatak 4

- ❑ Napišite funkciju koja prima polje cijelih brojeva te ih sortira od najmanjeg prema najvećem.
- ❑ Funkcija treba biti oblika:
 - `void sortiraj(int *polje, int N);`
- ❑ gdje je N broj elemenata polja.

Zadatak 5

- Napišite funkciju koja prima polje cijelih brojeva te vraća najmanji i najveći element.
- Funkcija treba biti oblika:
 - `void izracunajMinMax(int *polje, int N, int *min, int *max);`