

Napisati funkciju za ispis niza brojeva - demonstrira prenos nizova brojeva u funkciju.

```
#include <stdio.h>
void print_array(int a[], int n) {
    int i;
    for (i = 0; i < n; i++) printf("%d ",a[i]);
    putchar('\n');
    /* Obratite paznju na ovaj sadrzaj koji se stampa!!! */
    printf("sizeof(a) - u okviru fje : %d\n", sizeof(a));
}
main() {
    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
    printf("sizeof(a) - u okviru main : %d\n", sizeof(a));
    print_array(a, sizeof(a)/sizeof(int));
}
```

Pokazivaci - osnovni pojam

```
#include <stdio.h>
main() {
    int x = 3;
    int *px;
    printf("Adresa promenljive x je : %p\n", &x);
    printf("Vrednost promenljive x je : %d\n", x);
    px = &x;
    printf("Vrednost promenljive px je (tj. px) : %p\n", px);
    printf("Vrednost promenljive na koju ukazuje px (tj. *px) je : %d\n", *px);
    *px = 6;
    printf("Vrednost promenljive na koju ukazuje px (tj. *px) je : %d\n", *px);
    printf("Vrednost promenljive x je : %d\n", x);
}
```

swap: Demonstracija prenosa argumenata preko pokazivaca

```
#include <stdio.h>
void swap_wrong(int x, int y) {
    int tmp;
    printf("x : %p\n", &x); printf("y : %p\n", &y);
    tmp = x;
    x = y;
    y = tmp;
}
void swap(int* px, int* py) {
    int tmp;
    printf("px = %p\n", px); printf("py = %p\n", py);
    tmp = *px;
    *px = *py;
    *py = tmp;
}
```

swap: Demonstracija prenosa argumenata preko pokazivaca

```
main()
{
    int x = 3, y = 5;
    printf("Adresa promenljive x je %p\n", &x);
    printf("Vrednost promenljive x je %d\n", x);
    printf("Adresa promenljive y je %p\n", &y);
    printf("Vrednost promenljive y je %d\n", y);
    swap_wrong(x, y);
    printf("Posle swap_wrong:\n");
    printf("Vrednost promenljive x je %d\n", x);
    printf("Vrednost promenljive y je %d\n", y);
    swap(&x, &y);
    printf("Posle swap:\n");
    printf("Vrednost promenljive x je %d\n", x);
    printf("Vrednost promenljive y je %d\n", y);
}
```

Napisati funkcija koja istovremeno vraća dve vrednosti - kolicnik i ostatak dva data broja.

```
#include <stdio.h>
void div_and_mod(int x, int y, int* div, int* mod) {
    printf("Kolicnik postavljam na adresu : %p\n", div);
    printf("Ostatak postavljam na adresu : %p\n", mod);
    *div = x / y;
    *mod = x % y;
}
main() {
    int div, mod;
    printf("Adresa promenljive div je %p\n", &div);
    printf("Adresa promenljive mod je %p\n", &mod);
    div_and_mod(5, 2, &div, &mod);
    printf("Vrednost promenljive div je %d\n", div);
    printf("Vrednost promenljive mod je %d\n", mod);
}
```

Pokazivacka aritmetika

```
#include <stdio.h>
main() {
    char s[] = "abcde"; int t[] = {1, 2, 3, 4, 5};
    char* ps = &s[0]; int* pt = &t[0];
    printf("ps = %p\n", ps);
    printf("ps+1 = %p\n", ps+1);
    printf("ps+2 = %p\n", ps+2);
    printf("ps-1 = %p\n", ps-1);
    printf("ps-2 = %p\n", ps-2);
    printf("pt = %p\n", pt);
    printf("pt+1 = %p\n", pt+1);
    printf("pt+2 = %p\n", pt+2);
    printf("pt-1 = %p\n", pt-1);
    printf("pt-2 = %p\n", pt-2);
}
```

Pokazivacka aritmetika

```
for (ps = s; *ps; ps++) putchar(*ps);
putchar('\n');

ps = &s[3];
printf("s = %p\n", s);
printf("ps = %p\n", ps);
printf("ps - s = %d\n", ps - s);
pt = &t[3];
printf("t = %p\n", t);
printf("pt = %p\n", pt);
printf("pt - t = %d\n", pt - t);
}
```