



# STRUKTURE PODATAKA I ALGORITMI 1

## VEŽBE 7

Tijana Ristović  
Aleksa Cerovina  
Filip Radovanović  
Đorđe Nedić



```
#include <stdio.h>
void print_array(int* pa, int n);
main()
{
    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int num_of_elements = sizeof(a)/sizeof(int);
    int* pa;
    printf("Niz a : %p\n", a);
    printf("Adresa prvog elementa niza a (&a[0]) : %p\n",
&a[0]);
    pa = a;
    printf("Pokazivac pa ukazuje na adresu : %p\n", pa);
    printf("a + 3 = %p\n", a + 3);
    printf("&a[3] = %p\n", &a[3]);
    printf("pa[5] = %d\n", pa[5]);    printf("(pa + 5) = %d\n",
*(pa+5));
```



```
printf("sizeof(a) = %ld\n", sizeof(a));  
printf("sizeof(pa) = %ld\n", sizeof(pa));  
print_array(a, num_of_elements);  
print_array(pa, num_of_elements);  
}  
  
void print_array(int* pa, int n)  
{  
    int i;  
    for (i = 0; i<n; i++)    printf("%d ", pa[i]);  
    putchar('\n');  
}
```



- Napisati program koji unosi niz proizvoljne dimenzije i nalazi najveći element.

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n, *a, i, max;
    printf("Unesi dimenziju niza : ");    scanf("%d", &n);
    a = (int*) malloc(n*sizeof(int));
    if (a == NULL) {
        printf("Greska : Nema dovoljno memorije!\n");
        return 1;
    }
    for (i = 0; i<n; i++) {
        printf("a[%d]=", i);    scanf("%d", &a[i]);
    }
    PRONALAZENJA MAKSIMUMA NIZA
    free(a);
    return 0;
}
```



- Napisati program koji unosi niz proizvoljne dimenzije i nalazi najveći element – Ver.2

```
#include <stdio.h>
#include <stdlib.h>
int* CreateIntArray(int n) {
    return (int*) malloc(n*sizeof(int));
}
int main() {
    int n, *a, i, max;
    printf("Unesi dimenziju niza : "); scanf("%d", &n);
    a = CreateIntArray(n);
    if (a == NULL) {
        printf("Greska : Nema dovoljno memorije!\n");
        return 1;
    }
    /* Nadalje a koristimo kao obican niz */
    free(a);
    return 0;
}
```



- Program demonstrira niz kome se veličina tokom rada povećava.

```
#include <stdio.h>
#include <stdlib.h>
#define KORAK 10
int main() {
    int* a = NULL;
    int duzina = 0, alocirano = 0;
    int n, i;
    do {
        printf("Unesi ceo broj (-1 za kraj): ");
        scanf("%d", &n);
        if (duzina == alocirano) {
            alocirano = alocirano + KORAK;
            a = realloc(a, alocirano*sizeof(int));
        }
        a[duzina++] = n;
    } while (n != -1);
}
```

```
printf("Uneto je %d brojeva. Alocirano je ukupno %d  
    bajtova\n", duzina, alocirano*sizeof(int));  
printf("Brojevi su : ");  
for (i = 0; i<duzina; i++)    printf("%d ", a[i]);  
free(a);  
return 0;  
}
```



- Program demonstrira funkciju **calloc** - funkcija inicijalizuje sadržaj memorije na 0.

```
#include <stdio.h>
#include <stdlib.h>
#define BR_ELEM 10
main()
{
    int *m, *c, i;
    m = malloc(BR_ELEM*sizeof(int));
    c = calloc(BR_ELEM, sizeof(int));
    for (i = 0; i<BR_ELEM; i++)
        printf("m[%d] = %d\n", i, m[i]);
    for (i = 0; i<BR_ELEM; i++)
        printf("c[%d] = %d\n", i, c[i]);
    free(m);
    free(c);
}
```





# VRAĆANJE NIZA IZ FUNKCIJE.

```
#include <stdio.h>
#include <stdlib.h>
int* KreirajNiz(int n) {
    return (int*)malloc(n*sizeof(int));
}
int* UnosNiza(int n) {
    int *x,i;
    x=KreirajNiz(n);
    for(i=0;i<n;i++) scanf("%d",&x[i]);
    return x;
}
main() {
    int i,n,*a;
    printf("Unesi dimenziju niza "); scanf("%d",&n);
    a=UnosNiza(n);
    for(i=0;i<n;i++) printf("%5d",a[i]);
    printf("\n");
    free(a);
}
```