



Napredni C kurs

Namenski računarski sistemi
Čas 2, 2021/2022

Nizovi

- Uređen skup podataka istog tipa, označen jednim imenom
- Podaci su smesteni kontinualno u memoriji
- Ime = adresa prvog elementa
- Pristup svakom od članova polja se ostvaruje pomoću jednog ili više indeksa, zavisno od dimenzije niza

```
int k[5] = {11,22,33,44,55};
```

	adresa →				
	11	22	33	44	55
indeks	0	1	2	3	4

k - ime označava početak niza (adresu)
k[i] - vrednost niza na odstojanju i

```
int a[3][4] =  
{1,2,3,4,5,6,7,8,9,10,11,12};  
...  
{{1,2,3,4},{5,6,7,8},{9,10,11,12}};
```

	adresa →			
Redovi	[0] [0]	[0] [1]	[0] [2]	[0] [3]
	[1] [0]	[1] [1]	[1] [2]	[1] [3]
	[2] [0]	[2] [1]	[2] [2]	[2] [3]

Kolone

Pokazivači

- Pokazivač je promenljiva čija je vrednost adresa podatka smeštenog u memoriji, izražena u byte-ima
- Pokazivači mogu adresirati
 - Promenljive
 - Funkcije
- Deklaracija pokazivača
 - `tip_pod *pnt;`
- `tip_pod` deklariše format i veličinu podatka na toj adresi
- Uvećanje (++) i smanjenje (--) pokazivača menja njegovu vrednost za `sizeof(tip_pod)`
- Isto važi i za sabiranje i oduzimanje pokazivača sa celobrojnom vrednošću
 - $[pnt \pm n] = [pnt] \pm n \times \text{sizeof}(\text{tip_pod})$

Pokazivači na promenljive

- Postavljanje pokazivača
 - `char *bp = &chr;`
 - `int *ip = NULL; (0L)`
- * - sadržaj promenljive tip_pod na adresi iz pokazivača
 - `char t = *bp;`
- `void *` deklariše neoznačen pokazivač
 - samo adresa – odgovara `unsigned char *`
 - bez warning-a se može puniti pokazivač proizvoljnog tipa
- Pokazivači se mogu eksplicitno cast-ovati kao i promenljive
 - `unsigned j = *(unsigned*) pnt`
- Pokazivač na pokazivač
 - `int **pnt; char **bp -> char *bp[];`
- Niz pokazivača
 - `int *pnt [20];`

Nizovi i pokazivači

- `element Niz[10];`
- `element *pokEl = Niz;`
- `Niz[3] == *(Niz + 3) == pokEl[3] == *(pokEl + 3)`
 - pristup trećem (četvrtom) elementu niza Niz
- `&Niz[5] == Niz + 5 == &pokEl[5] == pokEl + 5`
 - adresa petog (šestog) elementa niza Niz
- `(char*)Niz + 7 == (char*)pokEl + 7`
 - adresa sedmog (osmog) bajta zauzetog memorijskog prostora
- `*((char*)Niz + 9) == *((char*)pokEl + 9)`
 - pristup devetom (desetom) bajta zauzetog memorijskog prostora

Višedimenzionalni nizovi

Niz

← pokNiz

Definicije matrica:

- `char Niz[3][5];`
 - vrši se i zauzimanje memorijskog prostora
- `char **pokNiz = (char**)Niz;`
 - Ne vrši se zauzimanje memorijskog prostora, već se samo postavlja pokazivač na početak
- `pokNiz[0]` - pokazivač na `char`

Regularna sekvenca za definisanje:

```
enum{DIM1=3, DIM2=5};  
char Niz[DIM1][DIM2];  
char *pokNiz    = (char*)Niz;
```

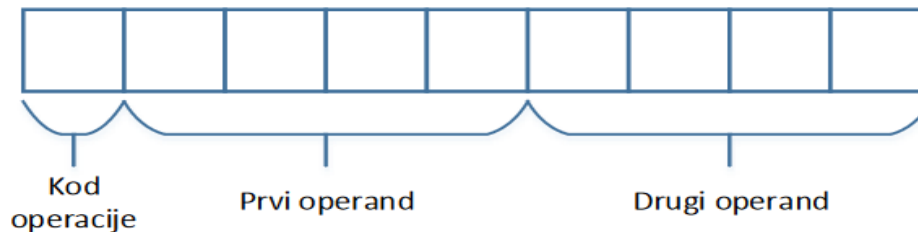
Linearizacija matrice (računanje adrese elementa matrice)

`Niz[2][3]` == `*(pokNiz + 2 * DIM2 + 3)`

Zadaci za vežbu

Zadatak 1:

- U programskom jeziku C/C++ implementirati funkciju čija je deklaracija:
`float* Calculate(char* buffer);`
- Funkcija kao parameter prima pokazivač na niz bajtova. Niz bajtova treba interpretirati kao na slici ispod. Prvi bajt predstavlja kod matematičke operacije (0 - sabiranje, 1 – oduzimanje, 2 – množenje, 3 - deljenje). Naredna četiri bajta predstavljaju prvi operand tipa integer, nakon čega slede naredna četiri bajta koja predstavljaju drugi operand tipa integer



- Na osnovu podataka dobijenih u nizu, primeniti odgovarajuću matematičku operaciju na operande. Povratna vrednost funkcije predstavlja pokazivač na broj u pokretnom zarezu (float) koji se dobije kao rezultat matematičke operacije.
- Napisati test program za ovu funkciju.

Zadaci za samostalnu vežbu

Zadatak 2:

- Napisati program koji učitava broj N ($N > 0$) tipa short i niz od N brojeva tipa integer, pakuje broj N i unete brojeve u niz bajtova i prosleđuje dobijeni niz funkciji average() koja izračunava i vraća najmanju i najveću vrednost u nizu. Zaglavlje fukcije average() dato je ispod

int* CalculateMinMax(char* buffer);

- Povratna vrednost fukcije je pokazivač na niz od dva podatka tipa integer koji predstavljaju najmanju i najveću vrednost u nizu.