

Nizovi

- ❖ Prvo pravilo: postoji implicitna konverzija niza elemenata tipa T u pokazivač na tip T , koja se vrši na svakom mestu gde se očekuje pokazivač, a pojavljuje se niz; vrednost ovog pokazivača ukazuje na prvi element datog niza (tzv. “rastakanje” niza u pokazivač, *array-to-pointer decay*):

$T[] \Rightarrow T^*$

Na primer:

f zapravo prima pokazivač na int

```
void f (int a[]) { cout<<a[0]<<endl; }
```

```
void g (int* p) { cout<<*p<<endl; }
```

```
int main () {
```

```
    int a[3] = {1, 2, 3};
```

```
    int* p = a;
```

```
    f(a);
```

```
    f(p);
```

```
    g(a);
```

```
    g(p);
```

```
}
```

Implicitna konverzija niza a tipa $int[3]$ u pokazivač na njegov prvi element, tipa int^* ; isto bi bilo za: $int^* p = \&a[0]$;

Implicitna konverzija niza a tipa $int[3]$ u pokazivač na njegov prvi element, tipa int^*

f zapravo prima pokazivač na int ; potpuno isti efekat kao i $f(a)$

Implicitna konverzija niza a tipa $int[3]$ u pokazivač na njegov prvi element, tipa int^*

Potpuno isti efekat kao i $g(a)$

Nizovi

- ❖ Drugo pravilo (tzv. pokazivačka aritmetika, *pointer arithmetics*): ako vrednost p tipa T^* ukazuje na element nekog niza elemenata tipa T , i ako je i vrednost celobrojnog tipa, definisana je operacija sabiranja i oduzimanja ove dve vrednosti; zbir je vrednost koja ukazuje na element istog niza koji je za i elemenata dalje od elementa na koju kazuje p (analogno za oduzimanje)
- ❖ Rezultat je validna vrednost pokazivača samo ukoliko p ukazuje na element niza i ako ovakvom aritmetikom vrednost rezultata ukazuje na element istog niza ili iza poslednjeg elementa niza; inače je vrednost invalidna
- ❖ Na primer:

```
int a[3] = {1, 2, 3};
int* p = &a[0];
int i;

for (i=0; i<3; i++, p++)
    -- *p;

++ *(p-1);
```

