

Nizi - primeri

Andrej Bagon

Deklaracija

Zgled deklaracije:

```
int main(void)
```

```
{
```

```
    char a[20]; /* niz je sestavljen iz 20 znakov */
```

```
    return 0;
```

```
}
```

Deklaracija

Inicializacija ob deklaraciji in izpis podatkov:

```
int main(void)
{
    char a[20] = "Primer"; /* niz je sestavljen iz 20
        znakov */
    printf("V spremenljivki a je vrednost %s\n",a);
    return 0;
}
```

Deklaracija

nepravilna uporaba (napako javi že prevajalnik!)

```
int main(void)
```

```
{
```

```
    char a[20];
```

```
    a="Primer"; // vrednosti prirejamo s pomočjo
```

```
        vgrajenih funkcij
```

```
    printf("%s\n",a);
```

```
    return 0;
```

```
}
```

Vnos podatkov

Branje podatkov:

```
int main(void)
{
    char a[20];
    printf("Vnesi niz: ");
    scanf("%s",a); /* ni znaka & pred spremenljivko*/
    printf("Vnesen niz je %s.\n",a);
    return 0;
}
```

Vnos podatkov

```
int main(void)
{
    char p[200];
    printf("Vnesi niz: ");
    scanf("%s",p); // pazi prebere samo eno besedo!
    printf("Vnesen niz je %s.\n",p);
    fflush(stdin);
    printf("Vnesi niz: ");
    fgets(p,200,stdin); // čaka na ENTER!! pazi '\n'
    printf("Vnesen niz je %s.\n",p);
    return 0;
}
```

Vnos podatkov

```
// zamenjamo skok v novo vrstico z ASCII  
znakom 0  
if (niz[strlen(niz)-1] == '\n')  
{  
    niz[strlen(niz)-1] = '\0';  
}
```

Standardne nizovne funkcije

- Uporaba knjižnice **string.h**
- Vnos podatkov
- Primerjava nizov med seboj
- Dolžina niza
- Združevanje nizov
- Pretvorba med števili in nizi, ter obratno

Standardne nizovne funkcije

- int strcmp(const char *string1, const char *string2)
- int strncmp(const char *string1, const char *string2, size_t n)
- char *strcpy(char *string1, const char *string2)
- char *strncpy(char *string1, const char *string2, size_t n)
- char *strcat(char *dest, const char *src)
- char *strncat(char *string1, const char *string2, size_t n)
- int strcasecmp(const char *s1, const char *s2)
- int strncasecmp(const char *s1, const char *s2, int n)
- int strlen(const char *string)
- char *strerror(int errnum)

Standardne nizovne funkcije

- char *strrev(char *s)
- char *strlwr(char *s);
- char *strupr(char *s);
- char *strchr(const char *string, int c)
- char *strrchr(const char *string, int c)
- char *strstr(const char *s1, const char *s2)
- char *itoa(int value, char *string, int radix)
- char *fcvt(double value, int ndig, int *dec, int *sign)
- int sprintf (char *buffer, const char *format [, argument, ...])

strcpy

```
#include <stdio.h>
#include <string.h>
int main(void)
{
    char a[20];
    strcpy(a,"Primer");
    printf("%s\n",a); // Izpiše: Primer
    return 0;
}
```

strcmp

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[20];
    strcpy(a,"Primer");
    printf("%d\n",strcmp(a,"ime")); // Izpise: -1 - p za i
    printf("%d\n",strcmp("ime",a)); // Izpise: 1 - i pred p
    printf("%d\n",strcmp(a,a));    // Izpise: 0 - niza sta enaka
    printf("%s\n",a);
    return 0;
}
```

strcat

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    char a[20];
    char b[20]=" za nize.";
    strcpy(a,"Primer");
    printf("%s\n",a);           // Izpise: Primer
    printf("%s\n",b);           // Izpise: za nize.
    printf("%s\n",strcat(a,b)); // Izpise: Primer za nize.
    printf("%s\n",a);           // Izpise: Primer za nize.
    return 0;
}
```

atoi in atof

```
#include <stdlib.h>
#include <stdio.h>
```

```
int main(void)
{
    int n;
    float m;
    char str[20] = "12345.67";

    n = atoi(str);
    m = atof(str);
    printf("string = %s int = %d in float = %0.2f\n", str, n, m);
    return 0;
}
```

itoa

```
#include <stdlib.h>
#include <stdio.h>
int main(void)
{
    int number = 12345;
    char string[25];

    itoa(number, string, 10);
    printf("int = %d string = %s\n", number, string);
    return 0;
}
```

sprintf

```
#include <stdlib.h>
#include <stdio.h>
int main(void)
{
    float stevilo = 6.5126;
    char str[256];
    sprintf(str, "%0.4f", stevilo);
    printf("%s\n", str); // izpise nam 6.5126
    return 0;
}
```



```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
void beri(char *niz) {
    printf("Vnesi niz: ");
    fgets(niz,256,stdin);
    if (niz[strlen(niz)-1] == '\n') {
        niz[strlen(niz)-1] = '\0';
    }
}

void izpis(const char *niz) {
    printf("%s",niz);
}

int main(void) {
    char s[256];
    beri(s);
    izpis(s);
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int i;
int t[26];
void nastavi(int *t) {
for (i=0;i<26;i++) t[i]=0;
}
void izpis(int *t) {
for (i=0;i<26;i++)
    if (t[i] > 0) printf("%c - %d\n",i+65, t[i]);
}
void kaj(int *t, const char *x) {
for (i=0; i < strlen(x); i++)
    t[x[i]-65]=t[x[i]-65]+1;
}
int main(void) {
nastavi(t); kaj(t,"PLAVALKA"); izpis(t);
nastavi(t); kaj(t,"TRST"); izpis(t);
return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void)
{
    char p[5][200];
    int i;
    for (i=0;i<5;i++)
    {
        printf("Vnesi niz: ");
        fgets(p[i],200,stdin);
    }
    printf("vnesel si naslednje nize:\n");
    for (i=0;i<5;i++)
        printf("%d.niz: %s\n",i,p[i]);
    return 0;
}
```

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
void beri(char *niz)
{
    printf("Vnesi niz: ");
    fgets(niz,256,stdin);
    if (niz[strlen(niz)-1] == '\n') { niz[strlen(niz)-1] = '\0'; }
}
void izpis(const char *niz) {
    printf("%s",niz);
}
int main(void) {
    char s[5][256];
    int i;
    for (i=0;i<3;i++) {
        beri(s[i]);
        izpis(s[i]); }
    return 0;
}
```