

# Vježbe za 6. *blitz*

PiPI 2005. Grupa 01  
Zdenko Šimić

# Teme za 6. blitz

---

## Nove cjeline:

- **Formati za unos** (`scanf`)
- **Formati za ispis** (`printf`)
- **Ugrađene funkcije iz** `string.h`
- **Ugrađene funkcije iz** `ctype.h`
- **Ulaz/izlaz** (`getchar`, `putchar`)
- **Ulaz/izlaz** (`gets`, `puts`)
- `typedef` (**bez strukture**)

## Ponovljene cjeline iz 5. blica:

- *void* funkcije
- *Call by reference* (bez polja)
- *Call by value* (primjeri gdje argument ne promijeni)
- Pokazivači  
(zasebno, u programu bez funkcije i aritmetike)
- Aritmetika s pokazivačima
- 1D polje kao argument funkcije
- 2D polje kao argument funkcije

# Pregled ugrađenih funkcija

---

## Standardne funkcije <stdio.h>

```
int getchar(void);  
int putchar(int ch);  
char * gets(char * str);  
int puts(const char * z);  
int printf(const char * format, arg1, ..., arg n);  
int scanf(const char * format, arg1, ..., arg n);
```

## Znakovne funkcije <ctype.h>

```
int isalpha(int ch);           int isalnum(int ch);  
int isdigit(int ch);  
int islower(int ch);           int isupper(int ch);  
int tolower(int ch);           int toupper(int ch);
```

## Funkcije znak. niza <string.h>

```
char * strlwr(char * str);  
char * strupr(char * str);  
int strlen(const char * str);  
char * strcat(char * str1, const char * str2);  
char * strcpy(char * str1, const char * str2);  
char * strncpy(char * str1, const char * str2, int n);  
int strcmp(const char * str1, const char * str2);  
char * strchr(const char * str, int ch);  
char * strstr(const char * str1, const char * str2);
```

# Formati za unos (**scanf**)

- **%s**
  - učitavanje u polje znakova sve do unosa praznine tab-a ili enter-a
- **%6s**
  - učitavanje u polje znakova najviše 6 znakova sve do unosa praznine tab-a ili enter-a
- **%[ samo]**
  - učitavanje u polje znakova sve do unosa znaka koji nije ' ', 's', 'a', 'm' 'o'
  - upisivanje tab-a i enter-a također znači kraj učitavanja
- **%[^NeTo]** ili **%[^N^e^T^o]**
  - učitavanje u polje znakova sve do unosa nekog od znakova 'N', 'e', 'T' ili 'o'
  - upisivanje praznine, tab-a i enter-a također znači kraj učitavanja
- **%c**
  - učitavanje jednog znaka
- **%d**
  - učitavanje cijelog broja sve do unosa praznine tab-a ili enter-a
- **%2d%3d**
  - učitavanje dva cijela broja jednog s dvije znamenke i drugog s tri znamenke, bez znaka između ili do unosa praznine tab-a ili enter-a
- **%o**
  - učitavanje oktalnog broja sve do unosa praznine tab-a ili enter-a
- **%x**
  - učitavanje heksadecimalnog broja sve do unosa praznine tab-a ili enter-a

# Formati za unos (scanf)

```
char slova[10];
```

```
scanf("%6s", slova);
```

Za unos: Primjer za  
slova="Primje"

```
scanf("%[ samo]", slova);
```

Za unos: Samo se  
slova="Samo s"

```
scanf("%[^NeTo]", slova);
```

Za unos: nEtONE  
slova="nEtO"

```
char slova[10];
```

```
int i, j;
```

```
scanf("%2d%3d", &i, &j);
```

Za unos: 987653  
i=98, j=765

```
scanf("%2d %s", &i, slova);
```

Za unos: 34 Neki unos  
i=34, slova="Neki"

```
scanf("%o", &i);
```

```
printf("x%X d%d o%o", i, i, i);
```

Za unos: 17  
x0F d15 o17

# Formati za ispis (printf)

## Ispis:

```
char at[] = "\n.....\n";
int m = 8, n=100, d = 0xa;
float x = 1.61803, y=-3.14159, q = 2.718;
char ac[20] = "NEKI TEKST", az[]="Znakovi";
printf("%s|%-3d %05.3f %.3s", at, m, x, &ac[5]);
printf("%s|n=%05d, y=%07.3f", at, n, y);
printf("%s|%4.1f %4.2f %4.0f", at, q, q, q);
printf("%s|Broj %.0f.", &at[0], q);
printf("%s|%-10s%-3X-%+3x%3d", at, az, d, d, d);
printf("%s|%-10s %04d %6.3f", at, "DA", 987, -y);
printf("%s|%5.2f%-6.4s", at, 10*x, "zlatni");
printf("%s|%-3d %03d", at, m, -m);
printf("%s|%03d%-5.1f", at, 2, 3.14);
printf("%s|%04d%4d%02d%2d", at, 4, 4, 20, 20);
printf("%s|%05.2f%5.3s\n", at, 1.2345, "Simbol");
```

```
.....
|8    1.618 TEK
.....
|n=00100, y=-03.142
.....
| 2.7 2.72    3
.....
|Broj 3.
.....
|Znakovi    A    -    a 10
.....
|DA                0987    3.142
.....
|16.18zlat
.....
|8    -08
.....
|0023.1
.....
|0004    42020
.....
|01.23    Sim
```

# Ugrađene funkcije iz `string.h`

```
#include <string.h>
...
char st[] = "neki tekst", *nz="znakovi-";
char slova[20];
strncpy(slova, st, 4);
printf("%s\n", slova);
slova[4]='\0';
printf("%s\n", slova);
printf("%s\n", strcpy(slova, nz));
printf("%s %d\n", slova+4, strlen(&slova[4]));
strcpy(slova, nz);
printf("%s\n", strstr(st, "tek"));
printf("%s\n", strstr(st, "t")+1);
printf("%c\n", *(strstr(st, "t")+2));
printf("%s\n", strcat(slova, st));
printf("%s\n", strchr(st, 'i'));
printf("%d\n", strcmp(st, nz));
printf("%d\n", strcmp("abc", "ABC"));
printf("%d\n", stricmp("abc", "ABC"));
printf("%s\n",strupr(st+5));
printf("%s\n",strupr(strncpy(&st[0], &nz[1], 4)));
```

## Ispis:

Nedefiniran ispis jer  
nema oznake kraja niza

neki

neki

znakovi-

ovi- 4

tekst

ekst

k

znakovi-neki tekst

i tekst

-1

1

0

TEKST

NAKO TEKST

# Ugrađene funkcije iz ctype.h

## Ispis:

```
#include <ctype.h>

int i=0, v=0, m=0;
char c='c', z='C', s='1';
char niz[]="Testiranje 123 ABCD";

printf("%d %d\n", isupper(z), isalpha(s));      1 0
printf("%d %c\n", isupper(c), toupper(c));      0 C
for(; niz[i]!='\0'; i++){
    if (islower(niz[i]) m++;
    v += isupper(niz[i]);
    niz[i] = toupper(niz[i]);
}                                                5 9
printf("%d %d\n", v, m);                        TESTIRANJE 123 ABCD
printf("%s\n", niz);                            99 c
printf("%d %c\n", tolower(z), tolower(z));      C
printf("%c\n", z);
```



# Ulaz/izlaz (getchar, putchar)

Ispis:

```
int i;
char niz[]="123 abc ABC";
char *ps = "pointer na string";

i=strlen(niz);
while(isupper(niz[--i])){
    putchar(niz[i]+32);
    if (niz[i]=='A') niz[i]='a';
    if (islower(niz[i])) break;
}
putchar('\n');
for(i=0; i<strlen(niz); i++)
    putchar(*(niz+i));
putchar('\n');
for (; *ps; ps++)
    if (*ps >= 'd' && *ps <= 'r')
        putchar(*(++ ps));
printf("\n%s", ps-4);
```

cba

123 abc aBC

onraig

ring

# Ulaz/izlaz (getchar, putchar)

```
void fprva(int i, char str[]) {
    for(; str[i]; i++)
        if (isdigit(str[i]))
            putchar(str[i]);
}

void fdruga (int i, char *str) {
    for (i=strlen(str)-1; i>=0; i--)
        putchar(*(str+i));
}

char *ftreca(int i, char *str) {
    do {
        str[++i] = getchar();
    } while( str[i] != 'X');
    str[i] = 0;
    return str;
}

void main () {
    char niz[]="\nT: ABC 123\n", as[40];
    fprva(0, niz);
    fdruga(0, niz);
    strcpy(as, ftreca(-1, niz));
    printf("%s %s\n", niz, as);
}
```

Ispis:

123

321 CBA :T

Unos s tastature:

test 1X↓

test 1 test 1

# Ulaz/izlaz (gets, puts)

---

## Rezultat/Ispis:

```
char niz[80], z;  
char str[] = "Neki tekst", *ps;  
  
puts(str);  
  
str[6]='\0';  
  
puts(str);  
  
puts("Unesi tekst: ");  
z = getchar();  
ps = gets(niz)+5;  
  
printf("%c%s\n", z, niz);  
printf("%s %d", ps, strlen(niz));
```

Neki tekst

Neki t

Unesi tekst:

jos jedan test

jos jedan test  
dan test 13

Uneseni  
tekst:

# typedef (bez strukture)

## typedef

- služi za definiranje korisničkih tipova podataka
- za ime korisničkog tipa nije dozvoljeno koristiti postojeće tipove podataka, ključne riječi niti velika slova
- novi tip se može definirati korištenjem prije definiranog tipa
- korisnički tip ne može biti novi tip podataka koji nije podržan u C-u
- zauzeće memorije za varijable definirane korisničkim tipom određeno je prema izvornom tipu podatka, npr.  

```
typedef long red;  
red a1[10];
```

polje `a1` zauzima 40 bajta
- korisnički tip može biti i pokazivač, npr.:  

```
typedef double *p2d;
```