

ZADACI

Definiranje pokazivača. Tipovi pokazivača. Korištenje pokazivača (inicijalizacija, pristup podatku, izmjena podatka)

1. Zadatak

Pažljivo proučite program:

```
#include <stdio.h>
int main () {
    int a = 3, *b;
    b = &a;
    scanf("%d", b);
    printf("%d\n", a);
    return 0;
}
```

i odgovorite što će se ispisati na ekran ako se na tipkovnici utipka **9** i Enter?

2. Zadatak

Što će se ispisati sljedećim programom?

```
#include <stdio.h>
int main() {
    int a = 2, b, *p;
    p = &a;
    b = *p;
    printf("a=%d b=%d\n", a, b);
    return 0;
}
```

Primjena pokazivača u dvodimenzionalnim poljima

1. zadatak

Učitajte vrijednosti za broj redaka (brRedaka) i broj stupaca (brStupaca) dvodimenzionalnog realnog polja, koje moraju biti manje ili jednake 10, a zatim učitajte članove tog polja. Potrebno je izračunati i ispisati umnožak svih članova matrice različitih od 0. Zadatak riješiti korištenjem pokazivača na dvodimenzionalno polje.

2. zadatak

Što će se ispisati izvođenjem sljedećeg programa?

```
#include <stdio.h>
#define MAX 5
int main() {
    float m[MAX][MAX] = {{1, 2}, {3, 4}};
    float *p = &m[0][0], umnozak = 1.;
    int i, j;

    for (i = 0; i < MAX; i++) {
        if (*(p + i * MAX + i) != 0) {
            umnozak *= *(p + i * MAX + i);
        }
    }
}
```

```

    }
    printf ("%f\n", umnozак);
    return 0;
}

```

Primjena pokazivača u jednodimenzionalnim poljima

1. zadatak

Učitajte vrijednost za broj elemenata (brElem) jednodimenzionalnog realnog polja, koje mora biti manje ili jednake 10, a zatim učitajte članove tog polja.

Potrebno je izračunati i ispisati omjer najvećeg i najmanjeg člana.

Npr. za polje 1, 7, -2, 9 omjer najvećeg i najmanjeg člana je $9/-2 = -4.5$.

Zadatak riješiti korištenjem pokazivača na jednodimenzionalno polje.

2. zadatak

Što će se ispisati izvođenjem sljedećeg programa?

```

#include <stdio.h>
int main(){

    int polje[] = {2,4,6,8};
    int *p = &polje[3];
    p -= 2;
    *p += *p;
    p++;
    printf("polje = %d %d %d %d \t*p=%d", polje[0], polje[1],
polje[2], polje[3], *p);
    return 0;
}

```

Definicija funkcije

1. zadatak

Definirati funkciju koja ne prima parametre, a uvijek vraća vrijednost 10 tipa int.

2. zadatak

Što je neispravno u definiciji funkcije nacetvrtu?

```

int nacetvrtu(int a){
    int a;
    return a*a*a*a;
}

```

Naredba return

1. zadatak

Koju vrijednost će vratiti funkcija ako ju pozovemo sa parametrom 2.7?

```
pv(2.7);
```

```
int pv(float a){
    float b;
    b=2*a;
    return;
}
```

2. zadatak

Koju vrijednost će vratiti funkcija ako ju pozovemo sa parametrom 2.7?

```
pv(2.7);
```

```
int pv(float a){
    float b;
    b=2*a;
    return b;
}
```

Void funkcije i funkcije bez argumenata

Što će ispisati sljedeći programi?

<u>1. zadatak</u>	<u>2. zadatak</u>
<pre>#include <stdio.h> void f(int a) { a*=a; printf("%d", a); } int main() { int a=3; f(a); printf("\n%d", a); return 0; }</pre>	<pre>#include <stdio.h> void f(int *a) { (*a)*=*a; printf("%d", *a); } int main() { int a=3; f(&a); printf("\n%d", a); return 0; }</pre>

Prijenos kopija vrijednosti (bez polja)

1. zadatak

Što će se ispisati obavljanjem sljedećeg programa?

```
char funkcija (int x, int y) {
    return (char) x + y;
}
```

```
int main () {
    int x = 40, y = 35;
    printf ("%d, %d, %c", x, y, funkcija (x, y));
    return 0;
}
```

2. zadatak

Što će se ispisati izvođenjem sljedećeg programskog odsječka:

```
float funkcija (float f) {
```

```

    if (f<0) return 2*f;
    return f;
}

int main() {
    int i=3, j=-3;
    printf("%f %f",funkcija(i), funkcija(j));
    return 0;
}

```

Prijenos predavanjem adrese argumenta (bez polja)

1. zadatak

Što će se ispisati sljedećim programom?

```

void f(int x, int *y) {
    x = 4;
    *y = x;
}

int main () {
    int a=2, b=3;
    f (a, &b);
    printf ("%d %d\n", a, b);
    return 0;
}

```

2. zadatak

Što će se ispisati na ekran sljedeći program?

```

void radi(char *x){
    char y = '1';
    *x = y;
}

int main(){
    char y = '3';
    radi(&y);
    printf("%c", y);
    return 0;
}

```

Jednodimenzionalna polja kao argumenti funkcije (rad s pokazivačima)

1. zadatak

Napisati funkciju *negativni* koja broji koliko elemenata u jednodimenzionalnom polju ima negativnu vrijednost. Napisati i poziv funkcije iz glavnog programa ?

```

#include <stdio.h>
int main () {
    int polje [25];
    int br_negativnih, i, n;
    printf ("Unesi broj elemenata i vrijednosti članova polja ");
}

```

```

scanf ("%d", &n);
for (i=0; i<n; i++) scanf ("%d", &polje[i]);
.../* prebroji negativne brojeve*/
return 0;
}

```

2. zadatak

Koji od sljedećih odgovora predstavlja ispravni prototip funkcije koja treba izračunati i vratiti najveći član u nizu?

- 1) void traziMax (long maxClan, int *niz, int brClan);
- 2) void traziMax (long *maxClan, int *niz, int brClan);
- 3) void traziMax (long maxClan, int niz, int brClan);
- 4) void traziMax (long *maxClan, int *niz[], int brClan);
- 5) void traziMax (long *maxClan, int niz, int brClan);

Smještajni razredi (postojanost, područje važenja varijabli) Samo elementarni pojmovi, po mogućnosti bez register i external!

Što će se ispisati u drugom pozivu funkcije f?

```

void f() {
    int a=3;
    static int b=2;
    b*=2;
    a+=b;
    printf("%d %d\n", b,a);
}

```

Dvodimenzionalna polje kao argument funkcije (rad s pokazivačima)

Ukoliko funkcija f treba izračunati sumu svih elemenata u matrici koju naredbu treba umetnuti na mjesto označeno s ### ?

```

int f(int *p, int m, int n, int maxstup){
    int i,j,suma=0;
    int *temp=p;
    for(i=0; i<m ; i++){
        for(j=0; j<n; j++){
            ###
        }
    }
    return suma;
}

```

- 1) suma += *temp + i * maxstup + j;
- 2) suma += temp + i * maxstup + j;
- 3) suma += *(temp[i * maxstup + j]);
- 4) suma += *(temp + i * maxstup + j);
- 5) suma += temp[i][j]

Jednodimenzionalna polja kao argumenti funkcije (rad s indeksnim izrazima)

Ukoliko se unutar funkcije u varijablu rez treba pridružiti umnožak elementa s indeksom retka 2 i indeksom stupca 4 iz matrice mat1 te elementa s indeksom retka 3 i indeksom stupca 5 iz matrice mat2 koje od sljedeći naredbi možemo navesti unutar sljedeće funkcije?

```
int f(int *mat1, int m, int n, int maxstup){
    int mat2[20][30];
    int *p = &mat2[0][0];
    ...
}
1) rez = mat1[2][4] * mat2[3][5];
2) rez = mat1[2*maxstup+4] * p[3*maxstup+5];
3) rez = mat1[2*maxstup+4] * mat2[3][5];
4) rez = mat1[2*maxstup+4] * p[3*30+5];
5) rez = mat1[2][4] * p[3*maxstup+5];
6) rez = mat1[2*maxstup+4] * p[3*20+5];
```

Macro s parametrima

Zadatak 1.

Što će se ispisati sljedećim programskim odsječkom?

```
#include <stdio.h>
#define m(x, y, z) (x) * (y-1) / z

int main() {
    int a = 3, b = 4, c = 5;
    printf("%d", m(++a, b%2, c-1));
    return 0;
}
```

Zadatak 2.

Napišite makro naredbu koja će utvrđivati je li zadano slovo malo slovo engleske abecede.

Matematičke ugrađene funkcije

Zadatak 1.

Kako bi se u C-u napisao sljedeći matematički izraz:

$$z = \frac{5}{2} \cdot \frac{\ln(x^3)}{\arccos(|y|) - 11}$$

Zadatak 2.

Kako bi se u C-u napisao sljedeći matematički izraz:

$$y = \frac{1}{2} \log(\sqrt{1-x^2})$$

Vlastite funkcije za rad s nizovima (deklariranim kao polje)

Zadatak 1.

Što će ispisati sljedeći program:

```
#include<stdio.h>
void f(char *p) {
    int i = 0;
    while (*(p+i) != '\0'){
        if (*(p+i) == ' ') printf("\n");
        else printf("%c", *(p+i));
        i++;
    }
}
int main () {
    char *p = "abc dabc ab";
    f(p);
    return 0;
}
```

Zadatak 2.

Napisati funkciju koja će za dani niz znakova (predan kao argument funkcije), vratiti broj znamenaka koje se pojavljuju u nizu.

Vlastite funkcije za rad s nizovima (deklariranim kao pokazivač)

Zadatak 1.

Što će ispisati sljedeći program:

```
#include<stdio.h>
void f(char p[], char c) {
    int i = 0;
    while (p[i] != '\0'){
        if (p[i] == c) p[i] = c - 32;
        i++;
    }
}
int main () {
    char p[20] = "abcdabcab";
    f(p, 'a');
    printf("%s", p);
    return 0;
}
```

Zadatak 2.

Napisati funkciju koja za zadani niz znakova (predan kao argument funkcije) i za zadani znak (također predan kao argument funkcije), vraća broj pojavljivanja znaka u nizu.

Aritmetika s pokazivačima

Zadatak 1.

Što će se ispisati sljedećim programskim odsječkom?

```
int polje[5]={5,4,3,2,1};
int i;
int *p;
p=&polje[2];
p++;
*p=*(p+1)-*(p-1);
for(i=0;i<5;i++)
{
    printf("%d ",polje[i]);
}
```

Zadatak 2.

Na koju adresu pokazuje pokazivač p nakon izvršavanja naredbe p=p+10; ?

Pokazivač je definiran naredbom int *p; , a prije naredbe je pokazivao na adresu 50000 u memoriji.

Ugrađene funkcije iz string.h

Zadatak 1.

Što će se ispisati sljedećim programskim odsječkom?

```
char niz[]="Vježbe za 6. blic";
char *p;
p=strchr(strchr(niz,'e'),'e');
printf("%s",p);
```

Zadatak 2.

Što će se ispisati sljedećim programskim odsječkom?

```
char niz[30]="Zadatak za vježbu";
char niz2[2]={'a','b'};
strcat(niz,niz2);
printf("%s ",niz);
```

Ugrađene funkcije iz ctype.h

Zadatak 1.

Što će ispisati sljedeći program?

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

int main() {
    char c1='z',c2='Z';

    if(isalnum(c1))
        if(isalpha(c2)) toupper(c1);
    else
        if(isalpha(c2)) tolower(c2);

    printf("%c %c",c1,c2);
    return 0;
}
```

Zadatak 2.

Koji je od sljedećih izraza ekvivalentan izrazu `if (iscntrl(c))`?

```
if(!isalpha(c) && !isdigit(c))
if(!isalpha(c) && !isdigit(c) && isprint(c))
if !(isalpha(c) || isdigit(c) || isprint(c))
if(!isalpha(c) && !isdigit(c) && !isspace(c))
if ((c >= 0 && c <= 31) || c==127)
```

Ulaz/izlaz (gets, puts)

Zadatak 1.

Napisati glavni program koji funkcijom `gets` (iz standardne biblioteke `string.h`) učitava niz znakova. Isti niz znakova, nakon učitavanja, treba funkcijom `puts` ispisati na zaslone.

Zadatak 2.

Napisati glavni program koji znakovne nizove s tipkovnice sve dok se ne unese prazan niz. Nizove je potrebno pohraniti u polje (napomena: koristiti dvodimenzionalno polje jer su znakovni nizovi sami po sebi polja; pretpostaviti da nizove neće biti više od 100 te da njihova duljina neće prelaziti 50 znakova). Nakon što su nizovi uneseni provjeriti koliko se puta unio niz „pero“.

Formati za ispis (printf)

Zadatak 1.

Što će se ispisati sljedećom naredbom:

```
printf("% d,% 4d,%-4d,%+4d,%04d", 12, -13, -14, 15, -16);
```

Zadatak 2.

```
printf("%8.2f %3.1f %-6.2f %.0f %.4f",
        1.2497f, 1235.14999f, 0.3542f, 3.72f, 1.2f);
```

Formati za unos (scanf)

Zadatak 1.

Što će ispisati sljedeći program:

```
#include <stdio.h>
int main(){
    int a,b;
    char c[11];
    scanf("%2d%2d%s", &a, &b, c);
    printf("%d %d %s", a, b, c);
    return 0;
}
```

ukoliko se kao ulaz s tipkovnice unese 1234567 89<ENTER>

Zadatak 2.

Što će ispisati sljedeći program:

```
#include <stdio.h>
int main(){
    int a, b, c;
    scanf("%o %x %d", &a, &b, &c);
    printf("%d %d %d", a, b, c);
    return 0;
}
```

ukoliko se kao ulaz s tipkovnice unese 12 12 12<ENTER>

Ulaz/izlaz (getchar, putchar)

Zadatak 1.

Što će se ispisati izvođenjem sljedećeg programa ?

```
#include <stdio.h>

int main() {
    char a[20] = "abcdeMNOP";
    int i = 0;
    while (a[i]) {
        putchar(tolower(a[i ++]));
    }
}
```

Zadatak 2.

Ako je pri izvođenju programa korisnik unio:

01234a<ENTER>

što se nakon toga ispisalo na zaslonu izvođenjem sljedećeg programa?

```
#include <stdio.h>
int main() {
    char a[20], znak;
    int i = 0;
    while (1) {
        znak = getchar();
        if (znak >= '0' && znak <= '9')
            a[i++] = znak;
        else
            break;
    }
    a[i] = 0;
    puts(a);
    return 0;
}
```

typedef (bez strukture)

Zadatak 1.

Koliko okteta u memoriji zauzima polje p:

```
typedef float decimal;
decimal p[10];
```

Zadatak 2.

Koja je od sljedećih naredbi **ispravna**?

1. typedef cijeliBroj int;
2. typedef int cijeliBroj;
3. typedef int = cijeliBroj;
4. typedef cijeliBroj = int;
5. typedef cijeliBroj(int);

RJEŠENJA

Definiranje pokazivača. Tipovi pokazivača. Korištenje pokazivača (inicijalizacija, pristup podatku, izmjena podatka)

1. Zadatak

Rješenje: 9

2. Zadatak

Rješenje: a=2 b=2

Primjena pokazivača u dvodimenzionalnim poljima

1. zadatak

```
#include <stdio.h>
#define MAXRED 10
#define MAXSTUP 10
int main() {
    float m[MAXRED][MAXSTUP];
    float *p = &m[0][0], umnozak = 1.;
    int brRedaka, brStupaca, i, j;

    printf("Upisite broj redaka i broj stupaca: ");
    scanf("%d %d", &brRedaka, &brStupaca);

    /* upis clanova polja */
    for (i = 0; i < brRedaka; i++) {
        for (j = 0; j < brStupaca; j++) {
            scanf("%f", p + i * MAXSTUP + j);
        }
    }

    // umnozak svih clanova polja */
    for (i = 0; i < brRedaka; i++) {
        for (j = 0; j < brStupaca; j++) {
            if (*(p + i * MAXSTUP + j) != 0) {
                umnozak *= *(p + i * MAXSTUP + j);
            }
        }
    }
    printf ("%f\n", umnozak);
    return 0;
}
```

2. zadatak

4.000000

Primjena pokazivača u jednodimenzionalnim poljima

1. zadatak

```
#include <stdio.h>

#define MAX 10

int main() {
    float polje[MAX];
    float *p = &polje[0];
    float max, min;
    int brElem, i;
    do {
        printf("Upisite broj elemenata: ");
        scanf("%d", &brElem);
    } while (brElem < 1 || brElem > 10);

    /* upis clanova polja */
    for (i = 0; i < brElem; i++) {
        scanf("%f", p + i);
    }

    min = max = *p;
    for (i = 1; i < brElem; i++) { /* primijetiti da petlja krece od
1 */
        if (min > *(p + i)){
            min = *(p + i);
        } else if (max < *(p + i)){
            max = *(p + i);
        }
    }
    /* Sto se dogadja ako je min=0? */
    printf ("Omjer je %f/%f=%f", max, min, max/min);
    return 0;
}
```

2. zadatak

polje = 2 8 6 8 *p=6

Definicija funkcije

1. zadatak

```
int funkcija()
{
    return 10;
}
```

2. zadatak

U funkciji se redefiniira formalni parametar a, što nije dozvoljeno!

Naredba return

1. zadatak

Funkcija će uvijek vratiti nedefiniranu vrijednost, bez obzira na vrijednost ulaznog parametra!

2. zadatak

5

Void funkcije i funkcije bez argumenata

1. zadatak

9

3

2. zadatak

9

9

Prijenos kopija vrijednosti (bez polja)

1. zadatak

40, 35, K

2. zadatak

3.200000 -6.400000

Prijenos predavanjem adrese argumenta (bez polja)

1. zadatak

2 4

2. zadatak

1

Jednodimenzionalna polja kao argumenti funkcije (rad s pokazivačima)

1. zadatak

```
int negativni (int *pom, int n) {  
    int i, br=0;  
    for (i=0; i<n; i++) if (pom[i]<0) br++;  
    return br;  
}
```

```
br_negativnih = negativni(polje, n);
```

2. zadatak

2

Smještajni razredi (postojanost, područje važenja varijabli) Samo elementarni pojmovi, po mogućnosti bez register i external!

Odgovor: 8 11

Dvodimenzionalna polje kao argument funkcije (rad s pokazivačima)

Odgovor: 4

Jednodimenzionalna polja kao argumenti funkcije (rad s indeksnim izrazima)

Odgovor: 3, 4

Macro s parametrima

Zadatak 1.

-1

Zadatak 2.

```
#define is_malo_slovo(x) ((x) >= 97 && (x) <= 122 ? 1: 0)
```

Matematičke ugrađene funkcije

Zadatak 1.

```
z = 5./2 * log(pow(x,3))/(acos(fabs(y))-11);  
ili  
z = 5./2 * log(x*x*x)/(acos(fabs(y))-11);
```

Zadatak 2.

```
y = 1. / 2 * log10(sqrt(1. - x*x) );  
ili  
y = 1. / 2 * log10(sqrt(1. - pow(x,2)) );
```

Vlastite funkcije za rad s nizovima (deklariranim kao polje)

Zadatak 1.

```
abc  
dabc  
ab
```

Zadatak 2.

```
int f(char p[]) {  
    int br = 0, i = 0;  
    while (p[i] != '\0'){  
        if (isdigit(p[i])) br++;  
        i++;  
    }  
    return br;  
}
```

Vlastite funkcije za rad s nizovima (deklariranim kao pokazivač)

Zadatak 1.

AbcdAbcAb

Zadatak 2.

```
int f(char *p, char c) {
    int br = 0, i = 0;
    while (*(p+i) != '\0') {
        if (*(p+i) == c) br++;
        i++;
    }
    return br;
}
```

Aritmetika s pokazivačima

Zadatak 1.

5 4 3 -2 1

Zadatak 2.

50040

Ugrađene funkcije iz string.h

Zadatak 1.

ezbe za 6. blic

Zadatak 2.

Ne može se točno odrediti jer niz nije terminiran '\0' znakom!

Ugrađene funkcije iz ctype.h

Zadatak 1.

z Z

Zadatak 2.

```
if ((c >= 0 && c <= 31) || c==127)
```

Ulaz/izlaz (gets, puts)

Zadatak 1.

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

int main()
{
    char niz[100];
    printf("Molim, unesite niz znakova: ");
    gets(niz);
    printf("Unijeli ste niz: %s", niz);
    return 0;
}
```

Zadatak 2.

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

int main()
{
    char niz[100][50];
    int ulazniBrojac = 0, brojacUsporedbe = 0;
    int i;

    do
    {
        //funkciji se daje adresa prvog znaka nekog retka polkja
        gets(&niz[ulazniBrojac++][0]);
        //provjerava se je li unesen prazan niz i je li
        //prijeđena gornja granica broja elemenata u polju
    } while((strcmp(&niz[ulazniBrojac-1][0], "") != 0)
        || ulazniBrojac >= 100);

    for(i = 0; i < ulazniBrojac-1; i++)
    {
        if(strcmp("pero", &niz[i][0]) == 0) brojacUsporedbe++;
    }

    printf("Niz \"pero\" se pojavio %d puta!\n", brojacUsporedbe);
}
```

Formati za ispis (printf)

Napomena: Znakom ^ su označene praznine

Zadatak 1.

$^12, ^{-13}, -14, ^^+15, -016$

Zadatak 2.

$^^^1.25^{1235.2^0.35}^{4^1.2000}$

Formati za unos (scanf)

Zadatak 1.

12 34 567

Zadatak 2.

10 18 12

Ulaz/izlaz (getchar, putchar)

Zadatak 1.

abcdemnop

Zadatak 2.

01234

typedef (bez strukture)

Zadatak 1.

40 okteta

Zadatak 2.

2