

ZADACI

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 referencija-adresa (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 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 ispravan 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];`

RJEŠENJA

Definicija funkcije

1. zadatak

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

2. zadatak

U funkciji se redefinira 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 referencija-adresa (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