

Pitanja i zadaci za vežbu

1.1 Koje su informacije pridružene pokazivaču? Koje od ovih informacija se u fazi izvršavanja čuvaju u memoriji dodeljenoj pokazivačkoj promenljivoj?

1.2. Koliko bajtova zauzima podatak tipa unsigned char? Koliko bajtova zauzima podatak tipa unsigned char*?

1.3. Ako je veličina koju zauzima element nekog tipa t 8 bajtova, koliko onda bajtova zauzima pokazivač na vrednost tipa t:

- (a) 4;
- (b) 8;
- (c) 16;
- (d) zavisi od sistema?

1.4. Ako je promenljiva tipa double * na konkretnom sistemu zauzima 4 bajta, koliko bajtova na istom sistemu zauzima promenljiva tipa unsigned char *?

1.5. Na šta se može primeniti operator referenciranja? Na šta se može primeniti operator dereferenciranja? Kako se označavaju ovi operatori? Šta je njihovo dejstvo? Kakav im je prioritet?

1.6. Ako je promenljiva p pokazivačkog tipa, da li je dozvoljeno koristiti izraz &p? Da li se umesto vrednosti p može pisati i *(&p)? Da li se umesto vrednosti p može pisati i &>(*p)?

1.7. Kako se označava generički pokazivački tip? Za šta se on koristi? Da li se pokazivač ovog tipa može referencirati? Zašto?

1.8. Ako je promenljiva tipa int*, da li joj se može dodeliti celobrojna vrednost? Ako je promenljiva tipa double*, da li joj se može dodeliti celobrojna vrednost?

1.9. Kog tipa je promenljiva a, a kog tipa promenljiva b nakon deklaracije short* a, b;? Koju vrednost ima promenljiva b nakon izvršavanja naredbi b = 2; a = &b; *a = 3; b++;?

2.1. Kako se prenose argumenti pokazivačkog tipa? šta to znači? Objasniti na primeru.

2.2. Šta ispisuju naredna dva programa?

```
#include<stdio.h>
int a = 1, b = 2;
void f(int* p) {
p = &b;
}
int main() {
int *p = &a
f(p);
printf("%d\n", *p);
}
```

```
#include<stdio.h>
void f(int* p) { p++; p += 2; p--; p = p + 1; }
int main() {
int a[] = { 1, 2, 3, 4, 5}, *p = a, *q = a;
f(p); printf("%d\n", *p);
q++; q += 2; q--; q = q + 1; printf("%d\n", *q);
}
```

2.3. Ukoliko je potrebno da funkcija vrati više od jedne vrednosti, kako se to može postići?

2.4. a) Napisati funkciju f sa povratnim tipom void koja ima samo jedan argument tipa int i vraća ga udvostručenog.

b) Napisati funkciju f sa povratnim tipom void koja ima tačno jedan argument, a koja prosleđenu celobrojnu vrednost vraća sa promenjenim znakom.

c) Napisati funkciju sa tipom povratne vrednosti void koja služi za izračunavanje zbira i razlike svoja prva dva argumenta.

Napisati program koji testira napisane funkcije.

2.5. Napisati funkciju koja za uneti broj sekundi ($0 \leq s < 86400$) proteklih od prethodne ponoći izračunava trenutno vreme tj. broj sati, minuta i sekundi.

3.1. Ako je u okviru funkcije deklarisan niz sa `char a[10]`; na koji deo memorije ukazuje `a+9`?

3.2. Ako je niz deklarisan sa `int a[10]`;, šta je vrednost izraza `a`?

Šta je vrednost izraza `a + 3`?

Šta je vrednost izraza `*(a+3)`?

3.3. Da li se komanda `ip = &a[0]` može zameniti komandom (odgovoriti za svaku pojedinačno)

(a) `ip = a[0]`;

(b) `ip = a`;

(c) `ip = *a`;

(d) `ip = *a[0]`?

3.4. Da li je vrednost `a[i]` ista što i (odgovoriti za svaku pojedinačno)

(a) `a[0]+i`;

(b) `a+i`;

(c) `*(a+i)`;

(d) `&(a+i)`?

3.5. Ukoliko je niz deklarisan na sledeći način: `float* x[10]`, kako se može dobiti adresa drugog elementa niza?

3.6. Neka je niz `a` deklarisan sa `int a[10]`, i neka je `p` pokazivač tipa `int*`. Da li je naredba `a = p`; ispravna? Da li je naredba `p = a`; ispravna? Koja je vrednost izraza `sizeof(p)` nakon naredbe `p = a`;, ako `int*` zauzima četiri bajta?

3.7. Niz je deklarisan sa `int a[10]`;. Da li je `a+3` ispravan izraz?

Da li mu je vrednost ista kao i vrednost `a[3]`, `&a[3]`, `*a[3]` ili `a[10]`?

Da li je `*(a+3)` ispravan izraz?

Da li mu je vrednost ista kao i vrednost `a[3]`, `a[10]`, `*a[3]` ili `*a[10]`?

3.8. Šta ispisuje naredni kod:

```
int a = 3, b[] = {8, 6, 4, 2}, *c = &a, *d = b + 2; *c = d[-1];
```

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

```
int a[] = {7, 8, 9}; int *p; p = a;
```

```
printf("%d %d\n", *(p+2), sizeof(p));
```

3.9. Ako je kao argument funkcije navedeno ime niza, šta se sve prenosi u funkciju:

(i) adresa početka niza;

(ii) elementi niza;

(iii) podatak o broju elemenata niza;

(iv) adresa kraja niza?

3.10. Koji prototip je ekvivalentan prototipu `int f(char s[])`?

Da li ima razlike između prototipova `void f(char *x)`; i `void f(char x[])`;

Da li ima razlike između deklaracija `char* x` i `char x[]`;

Da li su obe ispravne?

4.1. Ako je `p` tipa `int*`, šta je efekat naredbe `*p += 5`, a šta efekat naredbe `p += 5`?

4.2. Ako su promenljive `p` i `q` tipa `double *`, za koliko će se razlikovati vrednosti `p` i `q` nakon komande `p = q+n` (pri čemu je `n` celobrojna promenljiva)?

4.3. Ako je promenljiva p tipa double *, za koliko ce se promeniti njena vrednost

- (a) nakon komande ++*p;?
- (b) nakon komande ++(*p);?
- (c) nakon komande *p++;?
- (d) nakon komande (*p)++;?

Da li postoji razlika izmedju komandi (*p)++ i *(p++)?

4.4.

1. Koje binarne operacije su dozvoljene nad vrednostima istog pokazivačkog tipa (npr. nad promenljivima p1 i p2 tipa int*)?

2. Koje binarne aritmetičke operacije se mogu primeniti nad pokazivačem i celim brojem (npr. nad promenljivom p tipa int* i promenljivom n tipa int)?

4.5. Ako su p1, p2 pokazivači istog tipa, da li postoji razlika izmedju vrednosti: (p1-p2)+d i p1-(p2-d) ? Kog su tipa navedene vrednosti?

4.6. Ako su p1, p2, p3 pokazivači istog tipa, da li je naredna konstrukcija ispravna:

- (1) p=NULL;
- (2) p=10;
- (3) p1=p1+0;
- (4) p1=p1+10;
- (5) p1=(p1+p1)+0;
- (6) p1=(p1+p1)+10;
- (7) p1=(p1-p1)+0;
- (8) p1=(p1-p1)+10;
- (9) p1=(p1-p2)+10;
- (10) p1=p1-(p2+10);
- (11) p1=p1+(p1-p2);
- (12) p1=p1-(p1-p2);
- (13) p1=p1+(p2-p3);
- (14) p1=(p1+p2)-p3;
- (15) p1=(p1+p2)+p3;
- (16) p1=p1+(p2+p3);

4.7.

1. Za koliko bajtova se menja vrednost pokazivača p tipa int * nakon naredbe p += 2;?

2. Ako je p tipa float*, za koliko se razlikuju vrednosti p+5 i p?

3. Za koliko bajtova se menja vrednost pokazivača p tipa T * (gde je T neki konkretan tip), nakon naredbe p += 2;?

4. Ako je p pokazivač tipa T, čemu je jednako p + 3?

4.8.

1. Kog je tipa x, a kog tipa y nakon deklaracije int *x, *y;? Da li su dozvoljene operacije x+y i x-y?

2. Kog je tipa a, a kog tipa b nakon deklaracije int* a, b;? Da li su dozvoljene operacije a+b i a-b?

4.9. Da li nakon int n[] = { 1, 2, 3, 4, 5 }; int* p = n,;

sledece naredbe menjaju sadržaj niza n, vrednost pokazivača p, ili ni jedno ni drugo ili oba:

- (a) *(p++);
- (b) (*p)++;
- (c) *p++;?

6.1. Navesti primer inicijalizacije dvodimenzionog niza celih brojeva. Navesti primer deklaracije trodimenzionog niza realnih brojeva.

6.2. Da li se matrica a dimenzije 3*3 deklariše na sledeći način(odgovoriti za svaku pojedinačno):

- (a) a[3][3][3];
- (b) a[3,3,3];

- (c) a[3,3];
- (d) a[3][3];
- (e) a[3*3]?

6.3. Nakon deklaracije `int a[10]`; kog tipa je vrednost `a`?
Nakon deklaracije `int b[10][20]`; kog tipa je vrednost `b`?
Nakon deklaracije `int b[10][20]`; kog tipa je vrednost `b[0]`?

6.4. Objasniti naredne dve deklaracije i u čemu se razlikuju:
`char *name[] = { "Illegal month", "Jan", "Feb", "Mar" };`
`char aname[][15] = { "Illegal month", "Jan", "Feb", "Mar" };`

6.5. Kako se deklarirše trodimenzioni niz a dimenzija 10, 9, 8?

6.6. Koliko bajtova će biti rezervisano za naredne nizove:
`char *name[] = { "Illegal month", "Jan", "Feb", "Mar" };`
`char aname[][15] = { "Illegal month", "Jan", "Feb", "Mar" };`

6.7.

1. Da li je neposredno nakon deklaracije `char a[3][3]`, naredba `a[0][0] = 'a'`;
(a) sintaksno ispravna? (b) semantički ispravna?

2. Da li je neposredno nakon deklaracije `char* a[3]`, naredba `a[0][0] = 'a'`;
(a) sintaksno ispravna? (b) semantički ispravna?

6.8. Koliko bajtova zauzima i u kom delu memorije niz koji je u okviru neke funkcije deklarisan sa `char* a[10]`;

6.9. Koja inicijalizacija dvodimenzionog niza je ispravna:

- (a) `int a[][2] = {1,2,3,4}`;
- (b) `int a[2][] = {1,2,3,4}`;
- (c) `int a[][2] = {1,2,3,4}`;
- (d) `int a[2][] = {1,2,3,4}`;

6.10. Koliko je, na 32-bitnim sistemima, `sizeof(a)` i `sizeof(b)` nakon deklaracija `int* a[4]`; `int b[3][3]`?
Koliko je u opštem slučaju?

6.11. Navesti primer inicijalizacije niza pokazivača. Opisati efekat narednih deklaracija:
`int a[10][20]`; `int *b[10]`;

6.12. Pretpostavimo da se naredni kôd izvršava na 32-bitnom sistemu.

```
char* a[] = {"Dobar dan!", "Zdravo, zdravo"};  
char b[][15] = {"Dobar dan!", "Zdravo, zdravo"};  
Čemu je jednako sizeof(a) a čemu sizeof(b)?
```

6.13. Razmotriti naredni kod:

```
void f() {  
char* a = "Zdravo";  
char b[] = "Zdravo";  
...  
}
```

U kom segmentu memorije se nalazi promenljiva `a`?

U kom segmentu memorije je ono na šta pokazuje promenljiva `a`?

U kom segmentu memorije je ono na šta pokazuje promenljiva `b`?

6.14. Koliko bajtova će na steku biti rezervisano za niz

```
char *s[] = { "jedan", "dva", "tri" };
```

koji je lokalna promenljiva?

7.1. Ako se funkcija faktorijel sa prototipom `int faktorijel(int n)` može koristiti kao (jedini) argument funkcije `f` tipa `float`, kako je deklarirana funkcija `f`?

7.2. Da li je ispravan sledeći prototip funkcije čiji je prvi od dva argumenta pokazivač na funkciju koja ima jedan argument:

- (a) `int f(int* g(char), int x);`
- (b) `int f(void g(char), int x);`
- (c) `int f(int (*g)(char), int x);`
- (d) `int f(int (*g)(void), int x);`

7.3. Da li je ispravan sledeći prototip funkcije:
`int poredi(char *a, char *b, (*comp)(char *, char *));` ?
Obrazložiti odgovor.

7.4. Date su deklaracije:
`int *a1 (int* b1);`
`int (*a2) (int *b2);`
Kog je tipa `a1`, a kog `a2`?

7.5. Kog je tipa `x` deklarirano sa:

- 1. `double (*x[3])(int);`
- 2. `int (*x) (double);`
- 3. `int *x (double);`
- 4. `double* (*x) (float*);`
- 5. `int (*f) (float*);`

8.1. Navesti prototip, opisati njeno ponašanje i primer korišćenja funkcije:

- `_ malloc;`
- `_ calloc;`
- `_ realloc;`
- `_ free.`

U kom zaglavlju su deklarirane ove funkcije?

8.2. Funkcija `realloc` ima isti efekat kao funkcija `free` ako je:

- `_ vrednost njenog prvog argumenta nula;`
- `_ vrednost njenog drugog argumenta nula;`
- `_ vrednost oba njena argumenta nula;`
- `_ vrednost oba njena argumenta različita od nule.`

8.3. Kada je, nakon komandi

```
char *p;  
p = (char*)malloc(n);  
komanda strcpy(p, "test"); bezbedna?
```

8.4. U kom segmentu memorije se tokom izvršavanja programa čuvaju dinamički alocirani blokovi memorije.

8.5. Kako se zove pojava kada se izgubi vrednost poslednjeg pokazivača na neki dinamički alocirani blok memorije? Zašto je ova pojava opasna?

8.6. Da li će doći do curenja memorije ako nakon komande `p = (int*)malloc(sizeof(int)*5)` slede komanda/komande:

```
q = (int*)malloc(sizeof(int)*5);  
p = (int*)malloc(sizeof(int)*5);  
free(p); free(p);
```

```
free(p+1);
```

8.7. Zaokružiti komandu u narednom kodu koja dovodi do greške?

```
int *f = malloc(4*sizeof(int));
```

```
int* g = f;
```

```
free(g);
```

```
free(f);
```

```
f=g;
```

8.8. Koje su prednosti korišćenja dinamičke alokacije memorije (u odnosu na statičku i automatsku alokaciju)? Koji su nedostaci?

8.9. šta je, nakon poziva free(p), vrednost pokazivača p, a šta vrednost *p?

8.10. Koja naredba treba da se, prema dobroj praksi, izvrši nakon free(p);? šta se time postiže?

8.11. Kako sistem upravlja dinamičkom memorijom? Zašto se ne sme dinamički alociran prostor osloboditi dva puta?

8.12. Opisati ukratko bar četiri česte greške koje se javljaju u vezi sa dinamičkom alokacijom memorije.

8.13. Da li je u nekoj situaciji dozvoljeno funkciji free proslediti pokazivač koji nije dobijen funkcijama malloc, calloc i realloc?

8.14. Kako se zove situacija u kojoj je memorija na hipu podeljena na mnoštvo malih i nepovezanih blokova delova?

8.15. Da li se fragmentisanje memorije može umanjiti ako se?

_ ne koristi rekurzija;

_ ne koristi memorija na hipu;

_ koristi što manje lokalnih promenljivih;

_ koristi što manje globalnih promenljivih.

Zadatak 8.16. Napisati program koji sa standardnog ulaza učitava pozitivne cele brojeve dok ne učiata nulu kao oznaku za kraj. Na standardni izlaz ispisati koji broj se pojavio najviše puta medju tim brojevima. Na primer, ako se na ulazu pojave brojevi: 2 5 12 4 5 2 3 12 15 5 6 6 5 program treba da vrati broj 5. Zadatak rešiti korišćenjem dinamičke realokacije.

Zadatak 8.17. Napisati funkciju int f() koja učitava cele brojeve sa standardnog ulaza sve dok se ne učiata 0. Broj brojeva nije unapred poznat. Funkcija vraća broj brojeva većih od proseka. Ako nema unetih brojeva funkcija treba da vrati 0.

Zadatak 8.18. Napisati program koji sa standardnog ulaza unosi sortirani niz celih brojeva dok ne unesemo nulu kao oznaku za kraj (niz alocirati dinamički). U slučaju da niz nije sortiran ispisati podatak o grešci na standardni izlaz a u suprotnom ispisati medijanu tog niza. Medijana (sortiranog) niza koji ima n članova je, u slučaju kada je n neparan broj, središnji element niza odnosno, u slučaju kada je n paran broj, srednja vrednost dva središnja elementa. Na primer,

ako je dat niz 1, 3, 5, 6, 12, medijana je broj 5

a ako je dat niz 3, 5, 8, 13, 20, 25, medijana je broj 10:5.