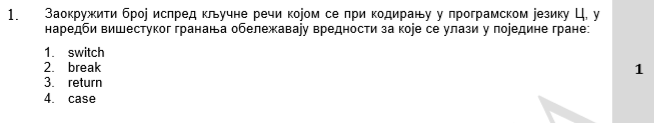
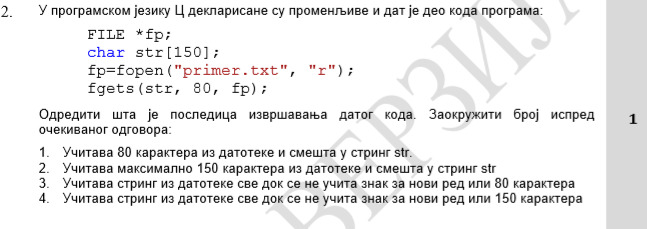
**Pitanja iz programskog jezika C ಠ\_ಠ**

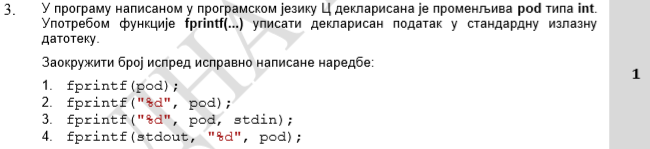
1-70



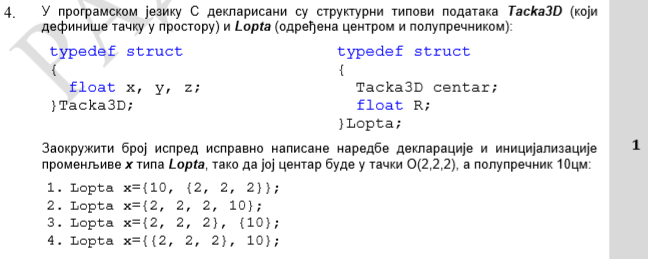
case



Учитава 80 карактера из датотеке и смешта у стринг str.



fprintf(stdout, "%d", pod);



Lopta x={2, 2, 2, 10};

Lopta x={{2, 2, 2}, 10};

Testirajte:

#include <stdio.h>

typedef struct {

float x, y, z;

}Tacka3D;

typedef struct {

Tacka3D centar;

float R;

}Lopta;

int main()

{

Lopta a={{2, 2, 2}, 10};

printf("%f %f %f",a.centar.x,a.centar.y,a.centar.y);

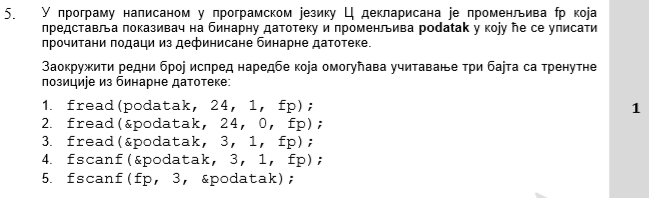
printf(" %f\n",a.R);

Lopta b={2, 2, 2, 10};

printf("%f %f %f",b.centar.x,b.centar.y,b.centar.y);

printf(" %f",b.R);

}



fread(&podatak, 3, 1, fp);

fread(void \*p, size\_t vel, size\_t n, FILE \*f)

Parametri:

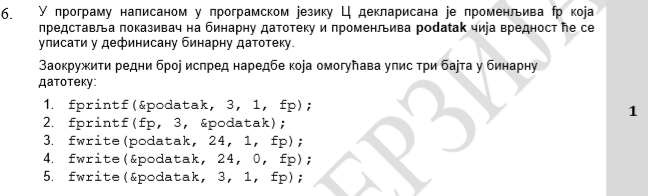
p – Ovo je pokazivač na blok memorije sa minimalnom veličinom vel\*n bajtova.

vel – Ovo je veličina u bajtovima svakog elementa koji se čita.

n – Ovo je broj elemenata, od kojih svaki ima veličinu bajtova veličine.

f – Ovo je pokazivač na objekat FILE koji specificira ulazni tok.

<https://overiq.com/c-programming-101/fread-function-in-c/>



fwrite(&podatak, 3, 1, fp);

fwrite(const void \*p, size\_t vel, size\_t n, FILE \*f)

Parametri

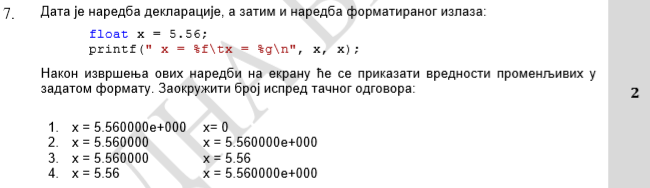
p – Ovo je pokazivač na niz elemenata za pisanje.

vel – Ovo je veličina u bajtovima svakog elementa koji će biti upisan.

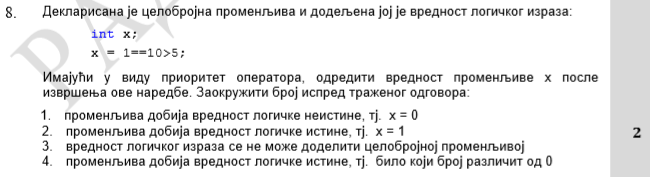
n – Ovo je broj elemenata, od kojih svaki ima veličinu bajtova veličine.

f – Ovo je pokazivač na objekat FILE koji specificira izlazni tok.

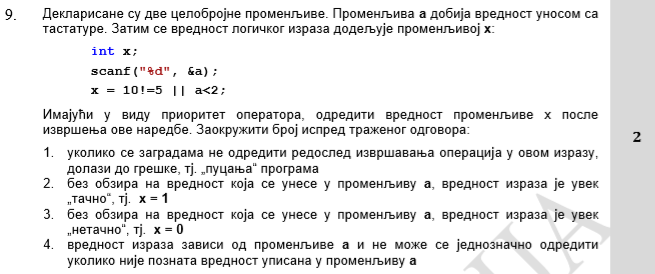
<https://overiq.com/c-programming-101/fwrite-function-in-c/>



x = 5.560000 x = 5.56

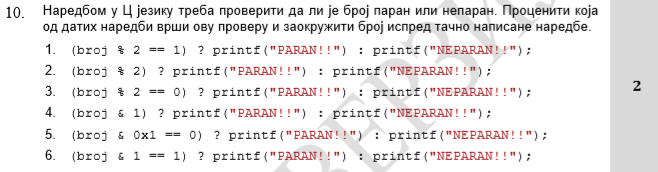


променљива добија вредност логичке истине, тј. x = 1

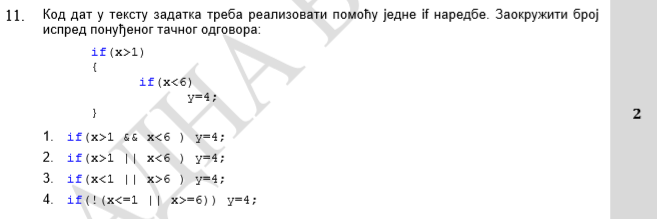


без обзира на вредност која се унесе у променљиву а, вредност израза је увек „тачно“,

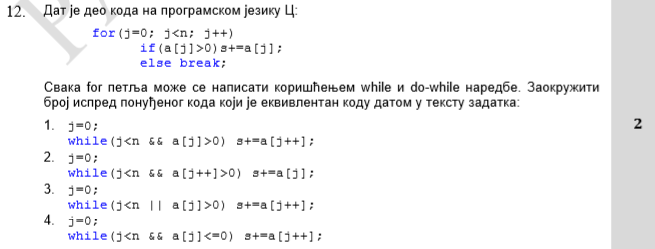
тј. x = 1



(broj % 2 == 0) ? printf("PARAN!!") : printf("NEPARAN!!");

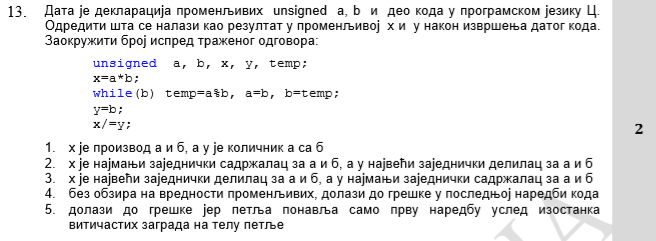


if(x>1 && x<6 ) y=4;

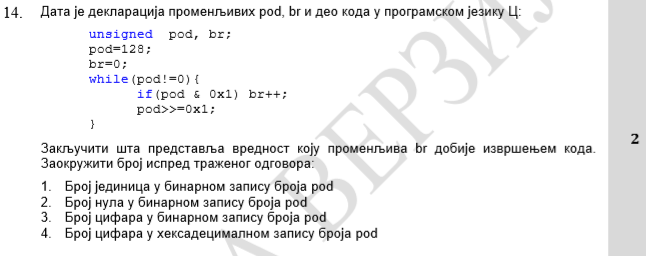


j=0;

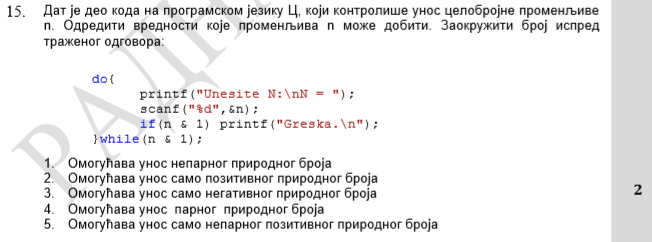
while(j<n && a[j]>0) s+=a[j++];



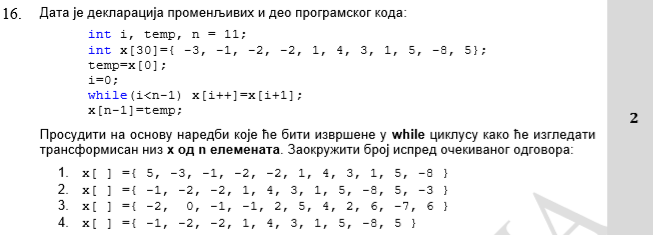
без обзира на вредности променљивих, долази до грешке у последњој наредби кода



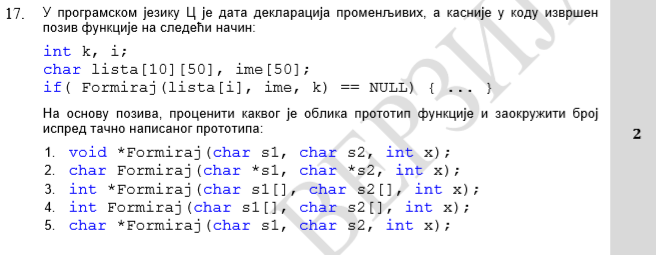
Број јединица у бинарном запису броја pod



Омогућава унос парног природног броја

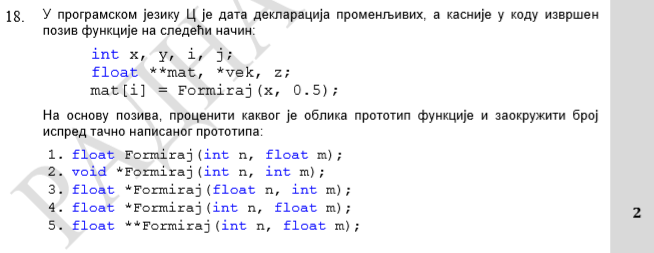


x[ ] ={ -1, -2, -2, 1, 4, 3, 1, 5, -8, 5, -3 }



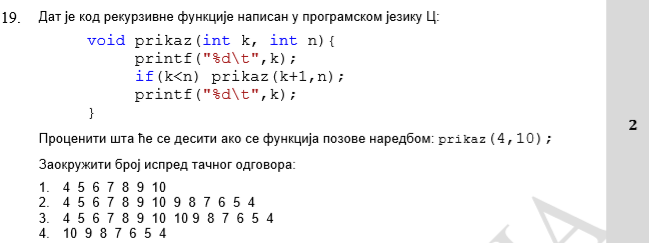
int \*Formiraj(char s1[], char s2[], int x);

**Iz if naredbe se vidi da rezultat funkcije mora da bude pokazivač, a parametri funkcije moraju da budu nizovi karaktera.**



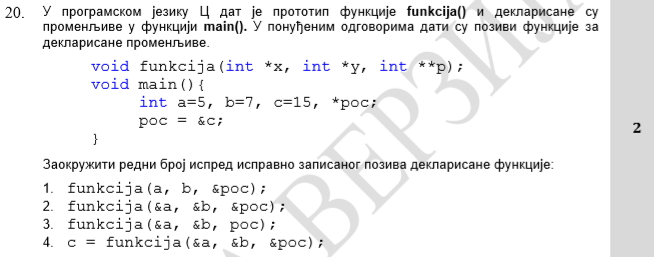
float \*Formiraj(int n, float m);

**Iz poziva funkcije se vidi da je rezultat funkcije pokazivač na niz realnih brojeva, a parametri moraju da budu ceo, pa realan broj.**

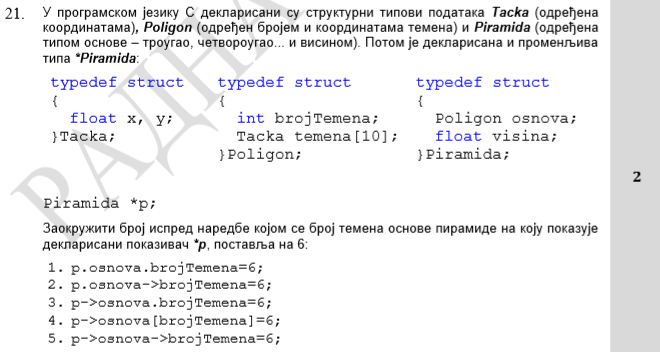


4 5 6 7 8 9 10 10 9 8 7 6 5 4

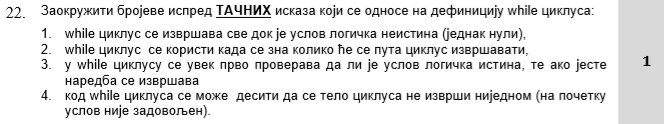
(zbog toga sto je rekurzivna funkcija u kojoj posle if-a imamo printf imamo ovo ponavljanje prethodnih vrednosti k od 10 do 4)



funkcija(&a, &b, &poc);

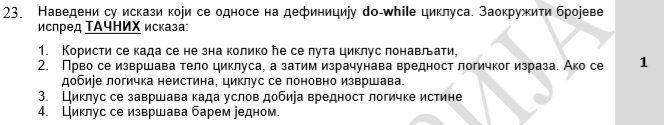


p->osnova.brojTemena=6;



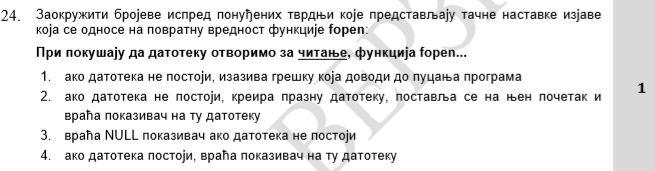
3. у while циклусу се увек прво проверава да ли је услов логичка истина, те ако јесте наредба се извршава

4. код while циклуса се може десити да се тело циклуса не изврши ниједном (на почетку услов није задовољен).



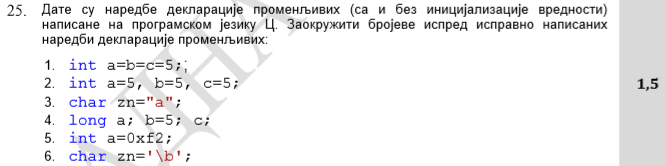
1. Користи се када се не зна колико ће се пута циклус понављати

4. Циклус се извршава барем једном



3. враћа NULL показивач ако датотека не постоји

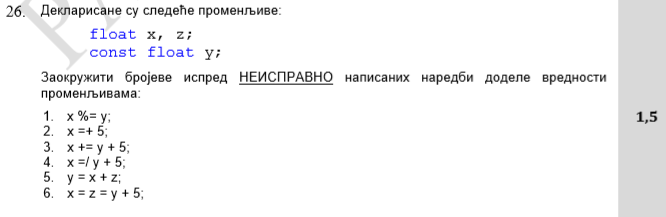
4. ако датотека постоји, враћа показивач на ту датотеку



int a=5, b=5, c=5;

int a=0xf2;

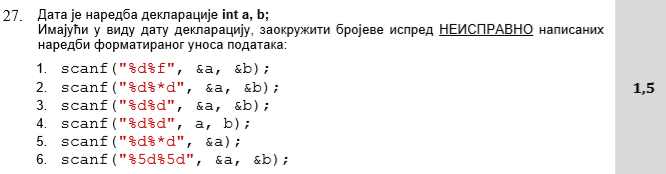
char zn='\b';



x %= y;

x =/ y + 5;

y = x + z;

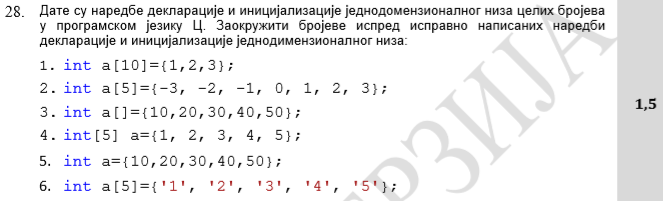


1. scanf("%d%f", &a, &b);

2. scanf("%d%\*d", &a, &b);

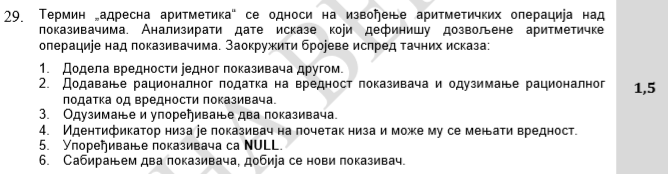
4. scanf("%d%d", a, b);

5. scanf("%d%\*d", &a);



int a[10]={1,2,3};

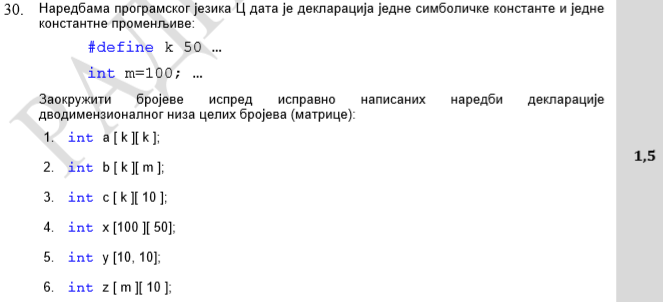
int a[]={10,20,30,40,50};



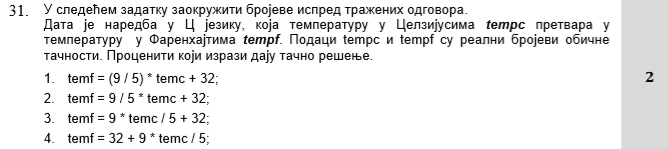
Додела вредности једног показивача другом.

Одузимање и упоређивање два показивача.

Упоређивање показивача са NULL.

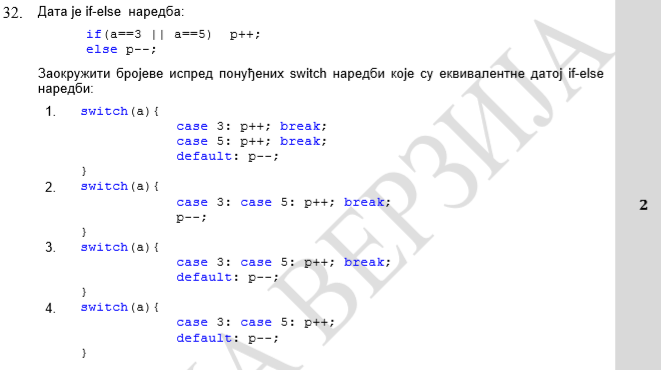


Tačan odgovor: 1, 2, 3, 4, 6

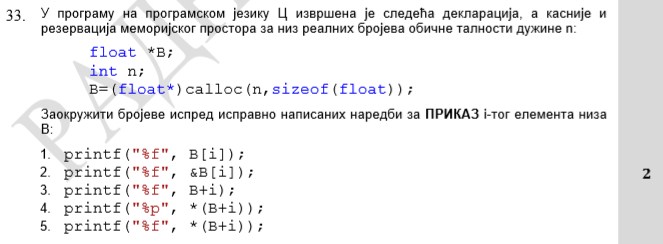


temf = 9 \* temc / 5 + 32;

temf = 32 + 9 \* temc / 5;



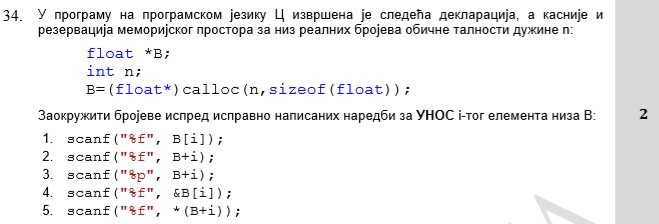
Tačan odgovor: 1, 3



printf("%f", B[i]);

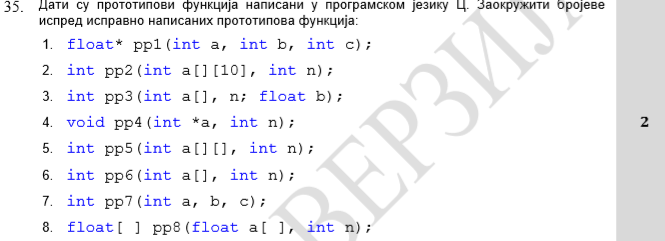
printf("%f", \*(B+i));

<https://docs.google.com/document/d/1JEtKOXgi8P5Id-NnnIanvjBCJnVZdBin/edit>

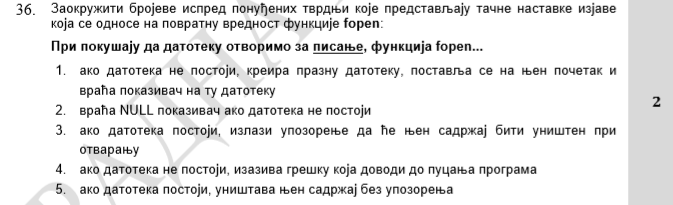


scanf("%f", B+i);

scanf("%f", &B[i]);



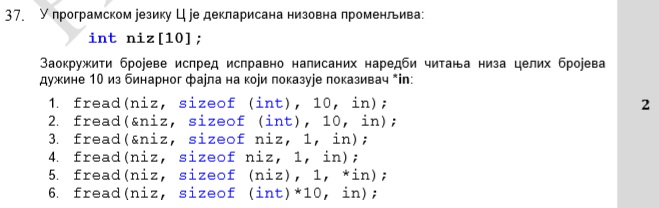
Tačan odgovor: 2, 4,5, 6



* ако датотека не постоји, креира празну датотеку, поставља се на њен почетак и

враћа показивач на ту датотеку 2

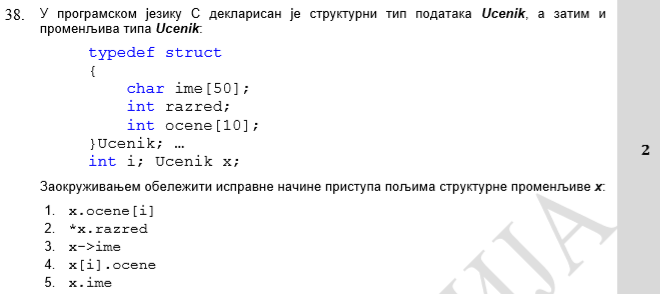
* ако датотека постоји, уништава њен садржај без упозорења



fread(niz, sizeof (int), 10, in);

fread(niz, sizeof niz, 1, in);

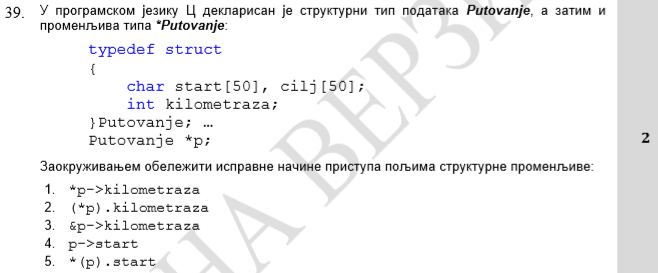
<https://overiq.com/c-programming-101/fread-function-in-c/>



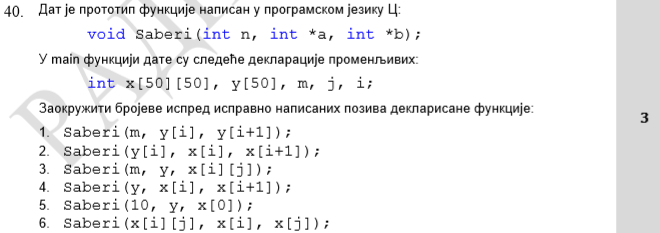
Tačan odgovor: 1, 5

x.ocene[i]

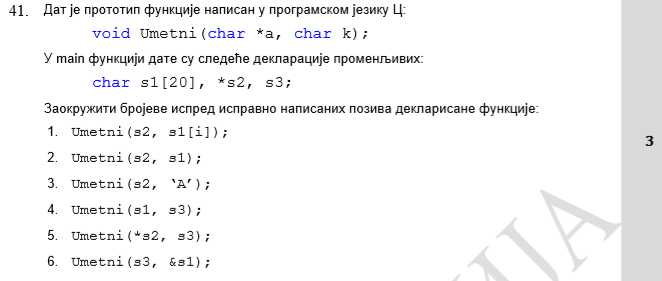
x.ime



(\*p).kilometraza



Saberi(m, y[i], y[i+1]);

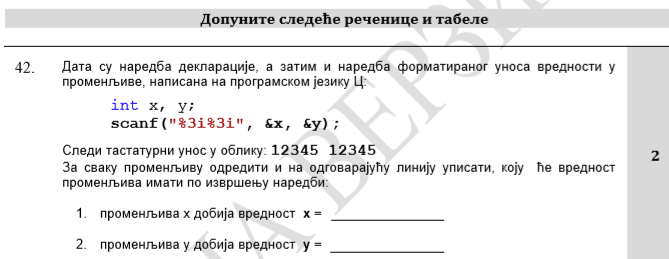


Tačan odgovor: 1, 3, 5

Umetni(s2, s1[i]);

Umetni(s2, ‘A’);

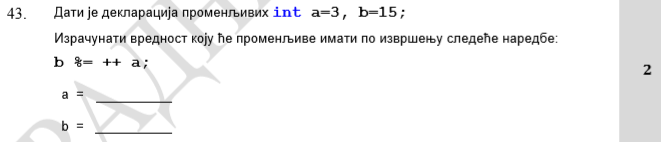
Umetni(\*s2, s3);



Rešenje:

1.promenljiva x добија вредност x=123

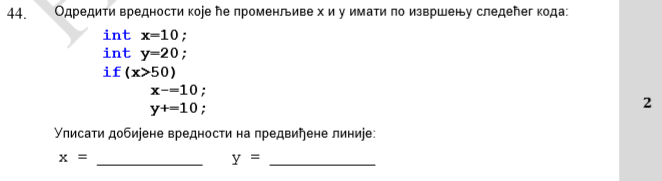
2.promenljiva y добија вредност y= 45



Rešenje:

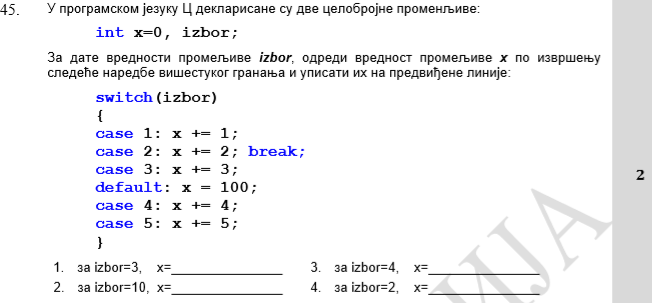
a= 4

b= 3



Rešenje:

x = 10 y = 30



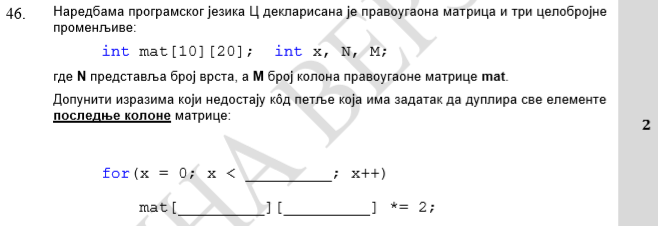
Rešenje:

за izbor=3, x = 109

за izbor=10, x = 109

за izbor=4, x = 9

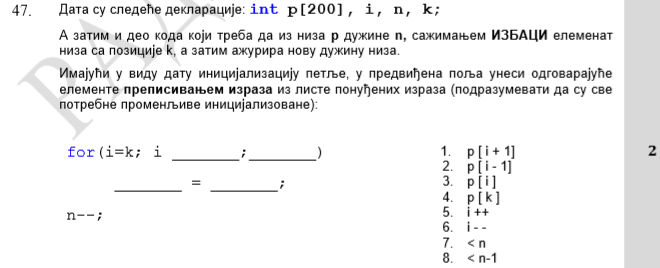
за izbor=2, x = 2



Rešenje:

for(x=0; x<N; x++)

mat[x][M-1]\*=2;

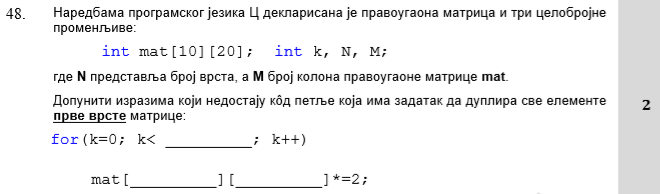


Odgovor:

for(i=k; i < n-1; i++ )

p[ i ] =p[ i +1] ;

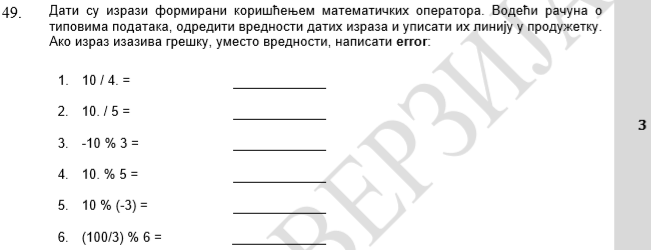
n--;



Odgovor:

for(k=0; k< M; k++)

mat[0,k]\*=2;



Odgovor:

10 / 4. = 2.5

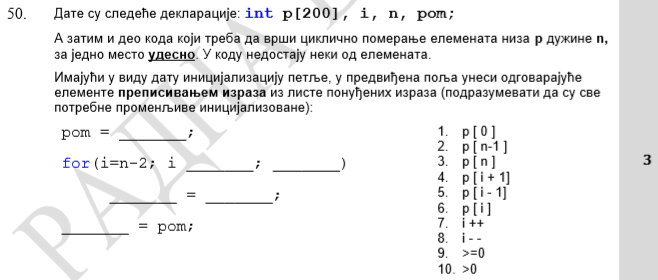
10. / 5 = 2.00000

-10 % 3 = -1

10. % 5 = error

10 % (-3) =1

(100/3) % 6 =3

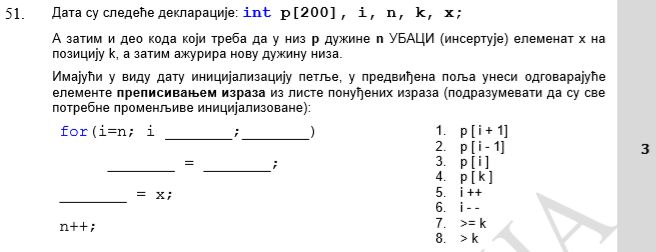


pom=p[n-1];

for(i=n-2; i >=0; i - -)

p[i+1]=p[i];

p[0]= pom;



1.odgovor:

for(i=n; i >=k; i - -)

p[i+1]=p[i];

p[k]= x;

n++;

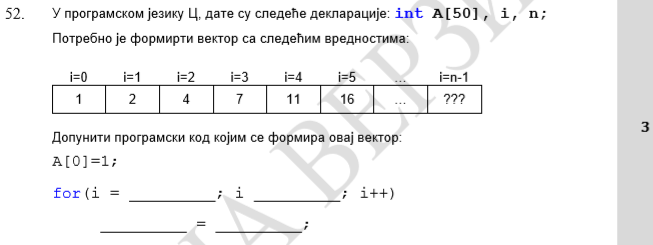
2.odgovor:

for(i=n; i >k; i - -)

p[i]=p[i-1];

p[k]= x;

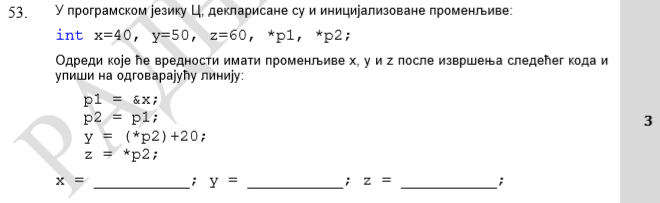
n++;



Odgovor:

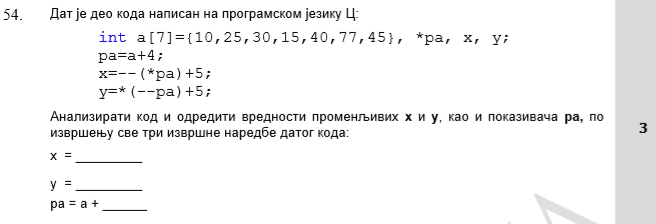
for(i=1;i<n;i++)

A[i]=A[i-1]+i;



Resenje:

x=40 y=60 z=40

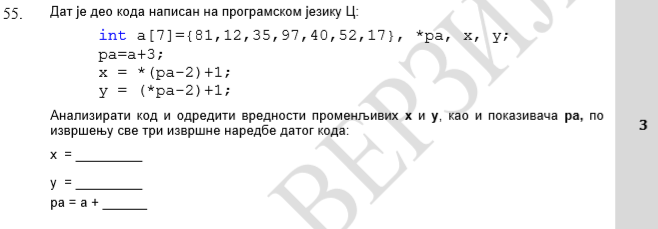


Odgovor:

x=44

y=20

pa=a+3

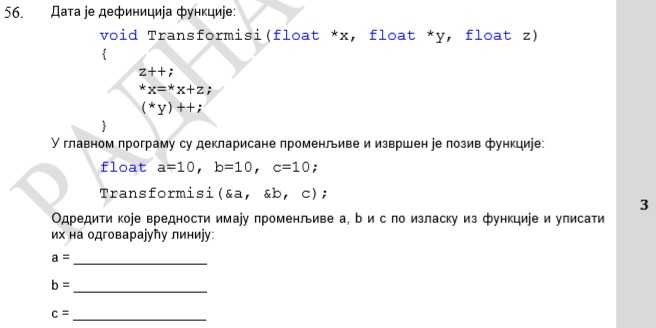


Odgovor:

x=13

y=96

pa=a+3

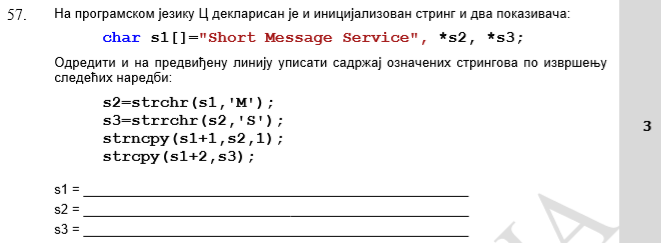


Resenje:

a= 21

b= 11

c= 10

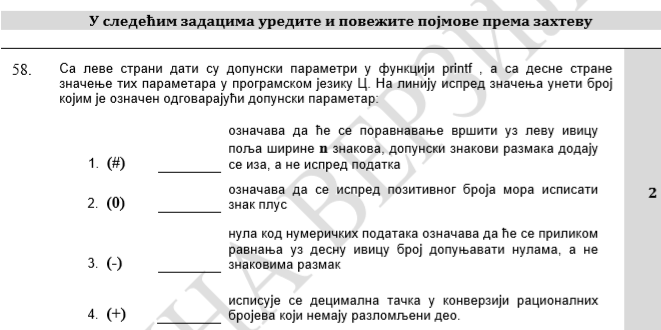


Rešenje:

String s1 je: SMService

String s2 je: ice

String s3 je: Service



Rešenje:

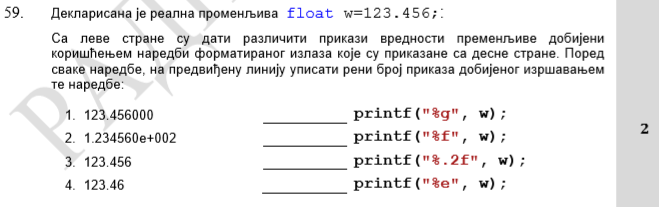
(-) означава да ће се поравнавање вршити уз леву ивицу поља ширине n знакова, допунски   
знакови размака додају се иза, а не испред податка

(+) означава да се испред позитивног броја мора исписати знак плус

(0) нула код нумеричких података означава да ће се приликом равнања уз десну ивицу број допуњавати нулама, а не знаковима размак

(#) исписује се децимална тачка у конверзији рационалних бројева који немају разломљени део.

<https://www.tutorialspoint.com/c_standard_library/c_function_printf.htm>



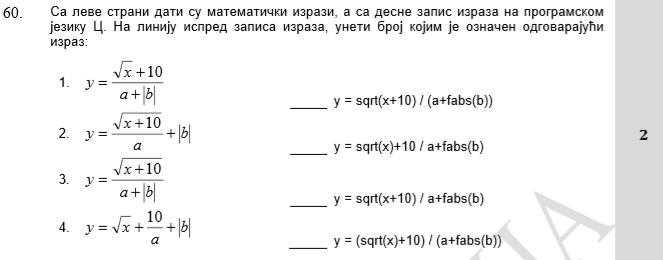
Rešenje:

123.456 printf(“%g”,w);

123.456000 printf(“%f”,w);

123.46 printf(“%.2f”,w);

1.234560e+002 printf(“%e”,w);



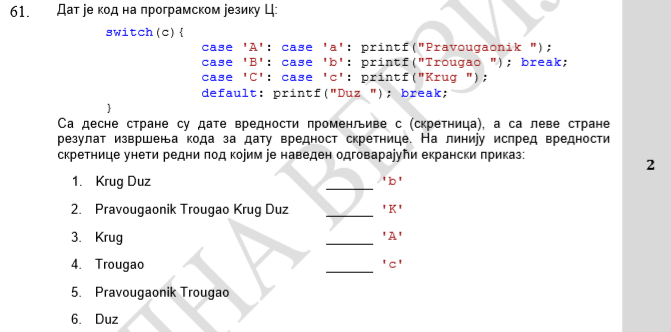
Resenje:

3.

4.

2.

1.



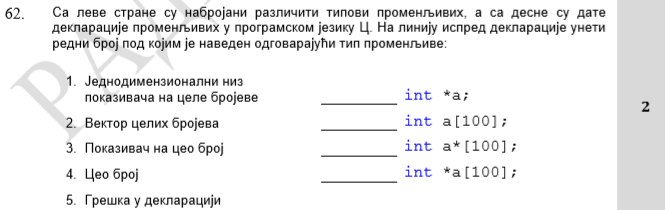
Resenje:

b trougao

K Duz

A pravougaonik trougao

c krug duz



Resenje:

int \*a; pokazivač na ceo broj

int a[100]; vektor (niz) celih brojeva

int a\*[100]; greška u deklaraciji

int \*a[100]; jednodimenzionalni niz pokazivača na cele objekte



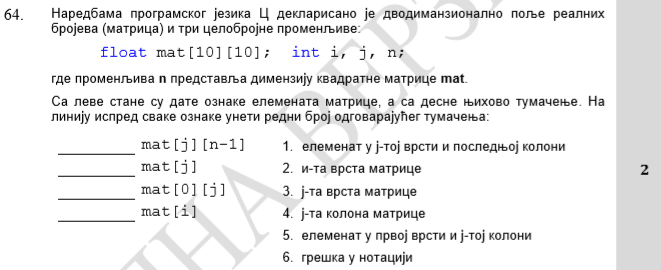
Resenje:

&a[0]; адреса почетног елемента низа

\*(a+n-1); вредност елемента на последњој позицији у низу

a+4; адреса петог елемента у низу

\*a; вредност елемента на почетној позицији у низу



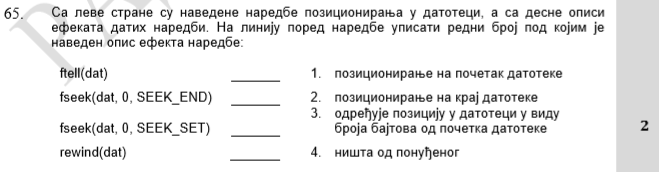
Resenje:

mat [ j ][ n-1 ] елеменат у ј-тој врсти и последњој колони

mat [ j ] грешка у нотацији

mat [ 0 ][ j ] елеменат у првој врсти и ј-тој колони

mat [ i ] грешка у нотацији



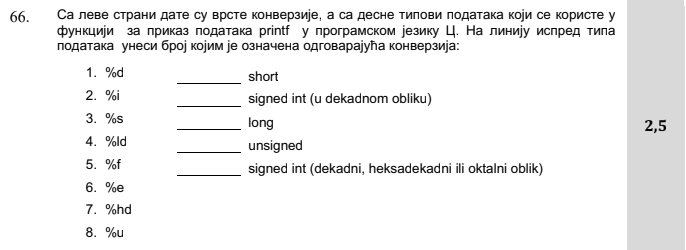
Resenje:

ftell(dat) одређује позицију у датотеци у виду броја бајтова од почетка датотеке

fseek(dat, 0, SEEK\_END) позиционирање на крај датотеке

fseek(dat, 0, SEEK\_SET) позиционирање на почетак датотеке

rewind(dat) ништа од понуђеног



Resenje:

7, 1 , 4, 8, 2

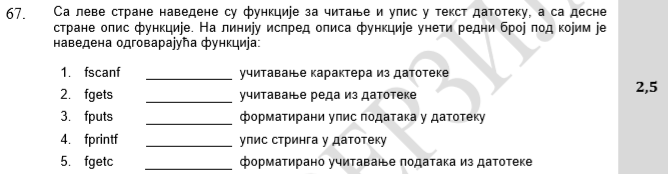
short %hd

signed int (u dekadnom obliku) %d

long %ld

unsigned %u

signed int (dekadni, heksadekadni ili oktalni oblik) %i



Resenje:

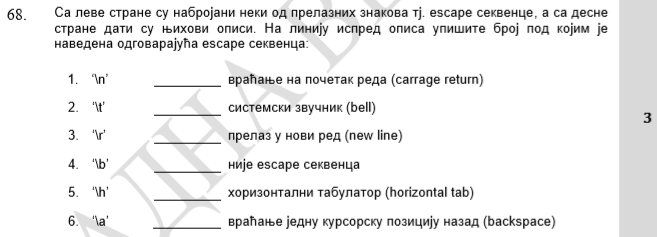
fgetc - učitavanje karaktera iz datoteke

fgets - učitavanje reda iz datoteke

fprintf - formatirani upis podataka u datoteku

fputs - upis stringa u datoteku

fscanf - formatirano učitavanje podataka iz datoteke



Resenje:

3 - vraćanje na početak reda (carrage return) **\r**

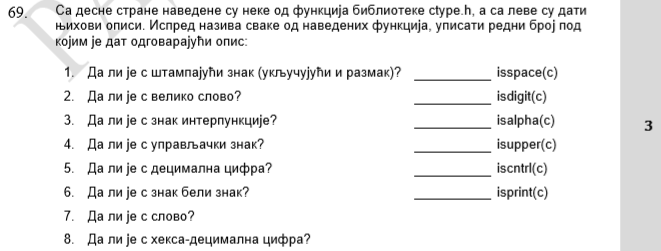
6 - sistemski zvučnik (bell) **\a**

1 - prelaz u novi red(new line) **\n**

5 - nije escape sekvenca **\h**

2 - horizontalni tabulator(horizontal tab) **\t**

4 - vraćanje jednu kursorsku poziciju nazad(backspace) **\b**



Resenje:

6 - isspace(c) da li je c znak beli znak

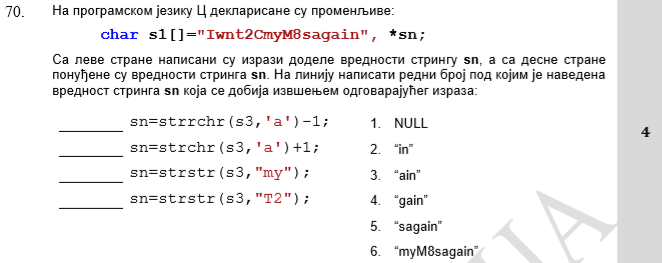
5 - isdigit(c) da li je c decimalna cifra

7 - isalpha(c) da li je c slovo

2 - isupper(c) da li je c veliko slovo

4 - iscntrl(c) da li je c upravljački znak

1 - isprint(c) da li je c štampajući znak



Resenje:

sn = strrchr(s1, 'a') - 1; **gain**

sn = strchr(s1, 'a') + 1; **gain**

sn = strstr(s1, 'my'); **myM8sagain**

sn = strstr(s1, "T2");  **NULL**

Funkcija char \*strrchr(const char \*str, int c) traži poslednju pojavu znaka c u nizu na koji ukazuje argument str.

Funkcija char \*strchr(const char \*str, int c) traži prvo pojavljivanje znaka c u nizu na koji ukazuje argument str.