

PITANJA ZA I KOLOKVIJUM

NIZOVI, STRINGOVI I POKAZIVAČI



NIZOVI, STRINGOVI I POKAZIVAČI

1. Funkcija koja prevodi znak ch u malo slovo je:

- **toupper(ch)**
- **tolower(ch)**



NIZOVI, STRINGOVI I POKAZIVAČI

1. Funkcija koja prevodi znak ch u malo slovo je:

- toupper(ch)
- 😊 tolower(ch)



NIZOVI, STRINGOVI I POKAZIVAČI

2. Niz po imenu `nums` koji sadrži 31 element tipa `short int` deklarirše se kao:

- `short int nums[31];`
- `nums short[31];`
- `short int nums(31);`
- `short nums[31];`



NIZOVI, STRINGOVI I POKAZIVAČI

2. Niz po imenu `nums` koji sadrži 31 element tipa `short int` deklarirše se kao:

- 😊 `short int nums[31];`
 - `nums short[31];`
 - `short int nums(31);`
- 😊 `short nums[31];`



NIZOVI, STRINGOVI I POKAZIVAČI

3. Ako definisem i inicijalizujem niz na sledeci nacin:

```
int niz[5] = {30, 50, 70, 90};
```

kolika ce biti vrednost elementa niz[4]?

- **0**
- **110**
- **nedefinisana (slucajan sadrzaj memorije)**



NIZOVI, STRINGOVI I POKAZIVAČI

3. Ako definisem i inicijalizujem niz na sledeci nacin:

```
int niz[5] = {30, 50, 70, 90};
```

kolika ce biti vrednost elementa niz[4]?



0

- 110
- **nedefinisana (slucajan sadrzaj memorije)**



NIZOVI, STRINGOVI I POKAZIVAČI

4. Niz se prosleđuje funkciji tako sto se svi elementi niza kopiraju u parametre.

- **Tacno**
- **Netacno**



NIZOVI, STRINGOVI I POKAZIVAČI

4. Niz se prosleđuje funkciji tako sto se svi elementi niza kopiraju u parametre.

- Tacno

 Netacno



NIZOVI, STRINGOVI I POKAZIVAČI

5. Funkcija `atof()` konvertuje numericki string u:

- **int**
- **float**
- **double**



NIZOVI, STRINGOVI I POKAZIVAČI

5. Funkcija `atof()` konvertuje numericki string u:

- `int`
- `float`
- 😊 `double`



NIZOVI, STRINGOVI I POKAZIVAČI

6. Pokazivac je promenljiva koja sadrzi _____ druge promenljive.

- **memorijsku adresu**
- **vrednost**



NIZOVI, STRINGOVI I POKAZIVAČI

6. Pokazivac je promenljiva koja sadrzi _____ druge promenljive.



- memorijsku adresu
• vrednost



NIZOVI, STRINGOVI I POKAZIVAČI

7. Ako je nums ime niza, onda se `nums[2]` preko pokazivača može napisati kao:

- `*(nums+2)`
- `&(nums+2)`
- `nums+2`



NIZOVI, STRINGOVI I POKAZIVAČI

7. Ako je nums ime niza, onda se `nums[2]` preko pokazivača može napisati kao:



`*(nums+2)`

- `&(nums+2)`
- `nums+2`

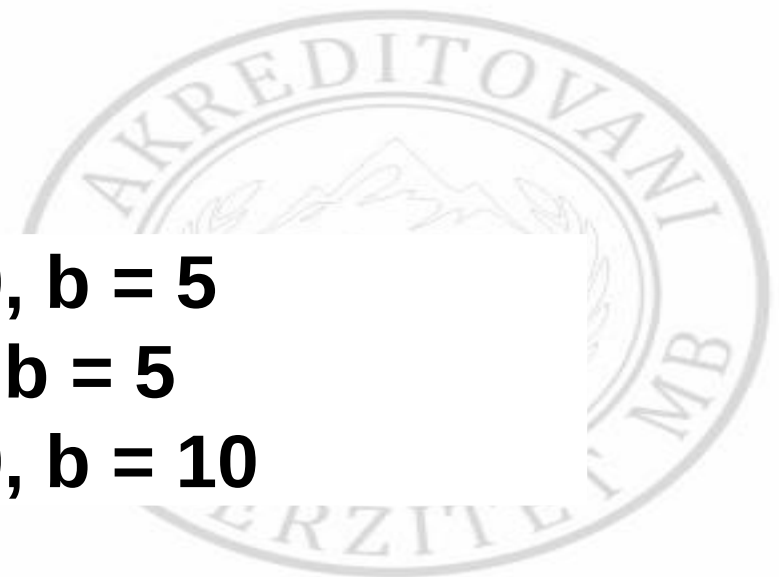


NIZOVI, STRINGOVI I POKAZIVAČI

8. Koje će vrednosti imati promenljive a i b nakon izvršavanja sledećeg koda:</br>

```
int a, b;  
int *p;  
a = 10;  
p = &b;  
*p = 5;  
p = &a;  
*p = a;
```

- **a = 10, b = 5**
- **a = 5, b = 5**
- **a = 10, b = 10**



NIZOVI, STRINGOVI I POKAZIVAČI

8. Koje će vrednosti imati promenljive a i b nakon izvršavanja sledećeg koda:</br>

```
int a, b;  
int *p;  
a = 10;  
p = &b;  
*p = 5;  
p = &a;  
*p = a;
```



a = 10, b = 5

- **a = 5, b = 5**
- **a = 10, b = 10**

NIZOVI, STRINGOVI I POKAZIVAČI

9. Koje će vrednosti imati promenljive a i b nakon izvršavanja sledećeg koda:


```
int a, b;
```

```
int *p;
```

```
a = 10;
```

```
p = &b;
```

```
*p = 5;
```

- **a = 10, b = 10**
- **a = 10, b = 5**
- **a = 3, b = 3**
- **a = 3, b = 10**



NIZOVI, STRINGOVI I POKAZIVAČI

9. Koje će vrednosti imati promenljive a i b nakon izvršavanja sledećeg koda:


```
int a, b;
```

```
int *p;
```

```
a = 10;
```

```
p = &b;
```

```
*p = 5;
```

- a = 10, b = 10
- 😊 a = 10, b = 5
- a = 3, b = 3
- a = 3, b = 10

NIZOVI, STRINGOVI I POKAZIVAČI

10. Dati reprezentaciju trećeg elementa u nizu koji se zove a.

- $a[2]$
- $a(2)$
- $a[3]$
- $a(3)$



NIZOVI, STRINGOVI I POKAZIVAČI

10. Dati reprezentaciju trećeg elementa u nizu koji se zove a.



- a[2]
- a(2)
 - a[3]
 - a(3)



NIZOVI, STRINGOVI I POKAZIVAČI

11. Koje su od sledećih izjava netačne?

- `int a[2];`
- `int a[];`
- `int a(2);`
- `int a = new int[2];`
- `int a() = new int[2];`



NIZOVI, STRINGOVI I POKAZIVAČI

11. Koje su od sledećih izjava netačne?

- `int a[2];`
-  `int a[];`
-  `int a(2);`
-  `int a = new int[2];`
-  `int a() = new int[2];`



NIZOVI, STRINGOVI I POKAZIVAČI

12. Ako se definiše niz `double list[] = {3.4, 2.0, 3.5, 5.5}`, `list[1]` je:

- **3.4**
- **2.0**
- **5.5**
- **undefined**



NIZOVI, STRINGOVI I POKAZIVAČI

12. Ako se definiše niz double `list[] = {3.4, 2.0, 3.5, 5.5}`, `list[1]` je:

- 3.4
- 😊 2.0
- 5.5
- undefined



NIZOVI, STRINGOVI I POKAZIVAČI

13. Ako se definiše niz `double list[] = {3.4, 2.0, 3.5, 5.5}`, najveći indeks u nizu `list` je:

- 0
- 1
- 2
- 3
- 4



NIZOVI, STRINGOVI I POKAZIVAČI

13. Ako se definiše niz `double list[] = {3.4, 2.0, 3.5, 5.5}`, najveći indeks u nizu `list` je:

- 0
- 1
- 2
- 3
- 4



NIZOVI, STRINGOVI I POKAZIVAČI

14. Koliko ima elemenata u nizu list[5]?

- 4
- 5
- 6
- 0



NIZOVI, STRINGOVI I POKAZIVAČI

14. Koliko ima elemenata u nizu list[5]?

- 4
- 😊 5
- 6
- 0



NIZOVI, STRINGOVI I POKAZIVAČI

15. Označiti tačne izjave u sledećem nizu izjava.

- **Svaki element u nizu je istog tipa.**
- **Veličina niza je fiksna nakon što se niz kreira.**
- **Veličina niza, koja se koristi pri definisanju niza, mora biti konstanta.**
- **Elementi niza su inicijalizirani kada se niz kreira.**

NIZOVI, STRINGOVI I POKAZIVAČI

15. Označiti tačne izjave u sledećem nizu izjava.

-  Svaki element u nizu je istog tipa.
-  Veličina niza je fiksna nakon što se niz kreira.
-  Veličina niza, koja se koristi pri definisanju niza, mora biti konstanta.
 - Elementi niza su inicijalizirani kada se niz kreira.

NIZOVI, STRINGOVI I POKAZIVAČI

16. Analizirati sledeći kod.

```
#include <iostream>
using namespace std;

int main()
{
    int x[3];
    cout << "x[0] je " << x[0];

    return 0;
}
```

- Program ima grešku prevođenja (engl. **compile error**), jer veličina niza nije navedena kod definisanja niza
- Program ima grešku u izvršavanju (engl. **runtime error**), jer elementi nisu inicijalizovani.
- Program se pokreće i prikazuje da je **x [0]** jednako 0.
- Program ima grešku u izvršavanju (engl. **runtime error**), jer element niza **x[0]** nije definisan.
- **x[0]** ima proizvoljnu vrednost.

NIZOVI, STRINGOVI I POKAZIVAČI

16. Analizirati sledeći kod.

```
#include <iostream>
using namespace std;

int main()
{
    int x[3];
    cout << "x[0] je " << x[0];

    return 0;
}
```

- Program ima grešku prevođenja (engl. **compile error**), jer veličina niza nije navedena kod definisanja niza
 - Program ima grešku u izvršavanju (engl. **runtime error**), jer elementi nisu inicijalizovani.
 - Program se pokreće i prikazuje da je x [0] jednako 0.
-  Program ima grešku u izvršavanju (engl. **runtime error**), jer element niza x[0] nije definisan.
-  x[0] ima proizvoljnu vrednost.

NIZOVI, STRINGOVI I POKAZIVAČI

17. Koje sledeće izjave su tačne?

- `int i(30);`
- `double d[30];`
- `int i[] = {3, 4, 3, 2};`
- `int[] i = {3, 4, 3, 2};`
- `int i[4] = {3, 4, 3, 2};`



NIZOVI, STRINGOVI I POKAZIVAČI

17. Koje sledeće izjave su tačne?

- `int i(30);`
- 😊 `double d[30];`
- 😊 `int i[] = {3, 4, 3, 2};`
- `int[] i = {3, 4, 3, 2};`
- 😊 `int i[4] = {3, 4, 3, 2};`



NIZOVI, STRINGOVI I POKAZIVAČI

18. Kako se inicijalizira niz sa dva karaktera tako da elementi niza budu 'a' i 'b'?

- `char[] charArray = {'a', 'b'};`
- `char[2] charArray = {'a', 'b'};`
- `char charArray[] = {'a', 'b'};`
- `char charArray[2] = {'a', 'b'};`



NIZOVI, STRINGOVI I POKAZIVAČI

18. Kako se inicijalizira niz sa dva karaktera tako da elementi niza budu 'a' i 'b'?

- `char[] charArray = {'a', 'b'};`
- `char[2] charArray = {'a', 'b'};`
- 😊 `char charArray[] = {'a', 'b'};`
- 😊 `char charArray[2] = {'a', 'b'};`



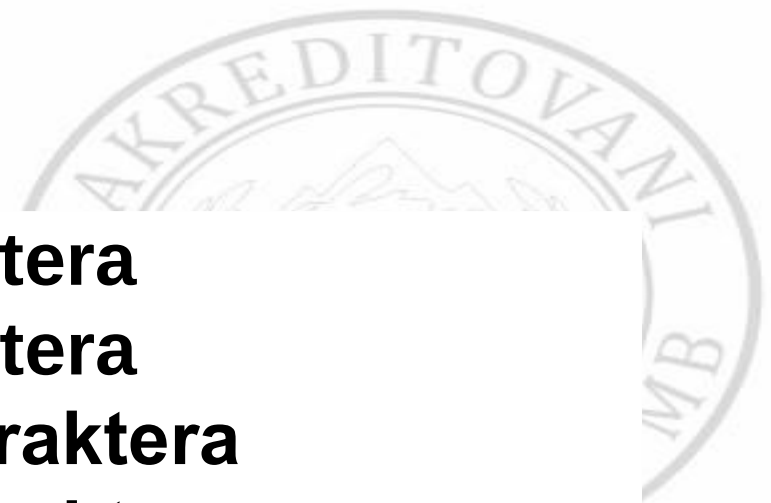
NIZOVI, STRINGOVI I POKAZIVAČI

19. Neka su data dva niza:

```
char s1[] = {'a', 'b', 'c'};  
char s2[] = "abc";
```

Označiti tačne izjave.

- **s1 ima tri karaktera**
- **s2 ima tri karaktera**
- **s1 ima četiri karaktera**
- **s2 ima četiri karaktera**



NIZOVI, STRINGOVI I POKAZIVAČI

19. Neka su data dva niza:

```
char s1[] = {'a', 'b', 'c'};  
char s2[] = "abc";
```

Označiti tačne izjave.

-  **s1 ima tri karaktera**
 - **s2 ima tri karaktera**
 - **s1 ima četiri karaktera**
-  **s2 ima četiri karaktera**

NIZOVI, STRINGOVI I POKAZIVAČI

20. Koja od sledećih izjava je tačna?

- `int count = 5; int* x = &count;`
- `int count = 5; int x = &count;`
- `int count = 5; int& x = &count;`
- `int count = 5; int** x = &count;`



NIZOVI, STRINGOVI I POKAZIVAČI

20. Koja od sledećih izjava je tačna?



- int count = 5; int* x = &count;**
- **int count = 5; int x = &count;**
 - **int count = 5; int& x = &count;**
 - **int count = 5; int** x = &count;**



NIZOVI, STRINGOVI I POKAZIVAČI

21. Ako je `int count = 5;`, koja od sledećih izjava je tačna?

- `&count` je adresa varijable `count`
- `&count` je 5
- `count` je adresa varijable `count`
- `count` is 5



NIZOVI, STRINGOVI I POKAZIVAČI

21. Ako je `int count = 5;`, koja od sledećih izjava je tačna?



&count je adresa varijable count

- **&count je 5**
- **count je adresa varijable count**
- **count is 5**



NIZOVI, STRINGOVI I POKAZIVAČI

22. Neka je `int count = 5` i `int* pCount = &count;` Koja sledeća izjava je tačna?

- `*count` je adresa za `count`
- `&count` je 5
- `*pCount` je 5
- `pCount` sadrži adresu za `count`



NIZOVI, STRINGOVI I POKAZIVAČI

22. Neka je `int count = 5` i `int* pCount = &count;` Koja sledeća izjava je tačna?

- `*count` je adresa za `count`
- `&count` je 5
- 😊 `*pCount` je 5
- 😊 `pCount` sadrži adresu za `count`



23. Znak (&) koji se koristi u sledećoj izjavi se zove:

```
int count = 5;  
cout << &count;
```

- **operator indirekcije (engl. indirection operator)**
- **operator dereferencije (engl. dereference operator)**
- **operator množenja**
- **operator adrese (engl. address operator)**

23. Znak (&) koji se koristi u sledećoj izjavi se zove:

```
int count = 5;  
cout << &count;
```

- operator indirekcije (engl. indirection operator)
- operator dereferencije (engl. dereference operator)
- operator množenja
- 😊 operator adrese (engl. address operator)

NIZOVI, STRINGOVI I POKAZIVAČI

24. Varijabla double je $d = 5.5$ i računar je čuva u memoriji na lokaciji 04BFA810. Onda je `&d` =

- 5
- 5.5
- 0
- nepoznata vrednost
- 04BFA810



NIZOVI, STRINGOVI I POKAZIVAČI

24. Varijabla double je $d = 5.5$ i računar je čuva u memoriji na lokaciji 04BFA810. Onda je $\&d =$

- 5
- 5.5
- 0
- nepoznata vrednost

 04BFA810



NIZOVI, STRINGOVI I POKAZIVAČI

25. Ispraviti greške u sledećem kodu.

```
int area = 1;
```

```
double* pArea = &area;
```

- greška se može ispraviti promenom `double* pArea = &area` u `double* pArea = area;`
- greška se može ispraviti promenom `double* pArea = &area` u `int* pArea = &area;`
- greška se može ispraviti promenom `double* pArea = &area` u `float* pArea = area;`
- greška se može ispraviti promenom `double* pArea = &area` u `int* pArea = area;`
- greška se može ispraviti promenom `int area = 1` u `double area = 1;`

NIZOVI, STRINGOVI I POKAZIVAČI

25. Ispraviti greške u sledećem kodu.

```
int area = 1;
```

```
double* pArea = &area;
```

- greška se može ispraviti promenom `double* pArea = &area` u `double* pArea = area;`



greška se može ispraviti promenom `double* pArea = &area` u `int* pArea = &area;`

- greška se može ispraviti promenom `double* pArea = &area` u `float* pArea = area;`
- greška se može ispraviti promenom `double* pArea = &area` u `int* pArea = area;`



greška se može ispraviti promenom `int area = 1` u `double area = 1;`

NIZOVI, STRINGOVI I POKAZIVAČI

26. Koja od sledećih izjava za definisanje novog tipa (sinonim ili drugo ime za postojeći tip) je tačna.

- **typedef postojećiTip noviTip;**
- **typedef noviTip postojećiTip**
- **typedef postojećiTip kao noviTip;**
- **typedef noviTip kao postojećiTip**

NIZOVI, STRINGOVI I POKAZIVAČI

26. Koja od sledećih izjava za definisanje novog tipa (sinonim ili drugo ime za postojeći tip) je tačna.



- typedef postojeciTip noviTip;**
- **typedef noviTip postojeciTip**
 - **typedef postojeciTip kao noviTip;**
 - **typedef noviTip kao postojeciTip**

NIZOVI, STRINGOVI I POKAZIVAČI

27. Ako se definiše niz `double list[] = {1, 3.4, 5.5, 3.5}` i računar to čuva na adresi 04BFA810, koje od sledećih izjava prikazuje 04BFA810?

- `cout << list << endl;`
- `cout << &list << endl;`
- `cout << list[0] << endl;`
- `cout << &list[0] << endl;`



NIZOVI, STRINGOVI I POKAZIVAČI

27. Ako se definiše niz `double list[] = {1, 3.4, 5.5, 3.5}` i računar to čuva na adresi 04BFA810, koje od sledećih izjava prikazuje 04BFA810?



`cout << list << endl;`



`cout << &list << endl;`



`cout << list[0] << endl;`



`cout << &list[0] << endl;`



NIZOVI, STRINGOVI I POKAZIVAČI

28. Neka je definisan niz `double list[] = {1, 3.4, 5.5, 3.5}`, a računar čuva taj niz na adresi `04BFA810` u memoriji. Pretpostaviti da tip `double` zauzima osam bajtova u memoriji računara. Onda je `&list[1]` je

- **04BFA810**
- **04BFA818**
- **1**
- **3.4**
- **nijedno od ponuđenih rešenja**

NIZOVI, STRINGOVI I POKAZIVAČI

28. Neka je definisan niz `double list[] = {1, 3.4, 5.5, 3.5}`, a računar čuva taj niz na adresi `04BFA810` u memoriji. Pretpostaviti da tip `double` zauzima osam bajtova u memoriji računara. Onda je `&list[1]` je

- **04BFA810**
- 😊 **04BFA818**
- **1**
- **3.4**
- **nijedno od ponuđenih rešenja**

NIZOVI, STRINGOVI I POKAZIVAČI

29. Neka je definisan niz double $\text{list}[] = \{1, 3.4, 5.5, 3.5\}$. $\&\text{list}[1]$ je isto što i:

- **list**
- **list + 1**
- **list + 2**
- **list[0]**
- **list[1]**



NIZOVI, STRINGOVI I POKAZIVAČI

29. Neka je definisan niz double `list[] = {1, 3.4, 5.5, 3.5}`. `&list[1]` je isto što i:

- `list`
- 😊 `list + 1`
- `list + 2`
- `list[0]`
- `list[1]`



NIZOVI, STRINGOVI I POKAZIVAČI

30. Neka je definisan niz double $list[] = \{1, 3.4, 5.5, 3.5\}$. $*(list + 1)$ je isto što i:

- $*list$
- $*list + 1$
- $*list + 2$
- $list[0]$
- $list[1]$



NIZOVI, STRINGOVI I POKAZIVAČI

30. Neka je definisan niz double `list[] = {1, 3.4, 5.5, 3.5}`. `*(list + 1)` je isto što i:

- `*list`
- `*list + 1`
- `*list + 2`
- `list[0]`
- 😊 `list[1]`



NIZOVI, STRINGOVI I POKAZIVAČI

31. Dati rezultat izvršavanja sledećeg koda:

```
#include <iostream>
using namespace std;
int main()
{
int list[] = {10, 20, 30, 40};
cout << *(list + 1) << " " << *list + 1 << endl;
return 0;
}
```

- **10 10**
- **20 20**
- **30 30**
- **20 11**
- **nijedno od ponuđenih rešenja**

NIZOVI, STRINGOVI I POKAZIVAČI

31. Dati rezultat izvršavanja sledećeg koda:

```
#include <iostream>
using namespace std;
int main()
{
int list[] = {10, 20, 30, 40};
cout << *(list + 1) << " " << *list + 1 << endl;
return 0;
}
```

- 10 10
- 20 20
- 30 30
- 😊 20 11
- nijedno od ponuđenih rešenja

NIZOVI, STRINGOVI I POKAZIVAČI

32. Dati rezultat izvršavanja sledećeg koda?

```
#include <iostream>
using namespace std;
int main()
{
    int list[] = {1, 1, 1, 1};
    *(list) = *(list) + 1;
    *(list + 1) = *(list + 1) + 2;
    *(list + 2) = *(list + 2) + 3;
    *(list + 3) = *(list + 3) + 4;
    cout << list[0] << " " << list[3] << endl;
    return 0;
}
```

- **1 2**
- **2 2**
- **3 4**
- **3 5**
- **2 5**

NIZOVI, STRINGOVI I POKAZIVAČI

32. Dati rezultat izvršavanja sledećeg koda?

```
#include <iostream>
using namespace std;
int main()
{
    int list[] = {1, 1, 1, 1};
    *(list) = *(list) + 1;
    *(list + 1) = *(list + 1) + 2;
    *(list + 2) = *(list + 2) + 3;
    *(list + 3) = *(list + 3) + 4;
    cout << list[0] << " " << list[3] << endl;
    return 0;
}
```

• 1 2

• 2 2

• 3 4

• 3 5



2 5

NIZOVI, STRINGOVI I POKAZIVAČI

33. Neka je dato:

```
int list1[4], list2[4];
```

```
int* p1; int* p2;
```

Koje su sledeće izjave tačne?

- **p1 = list1;**
- **p1 = p2;**
- **list1 = p1;**
- **list1 = list2;**
- **sve od datog**

NIZOVI, STRINGOVI I POKAZIVAČI

33. Neka je dato:

```
int list1[4], list2[4];
```

```
int* p1; int* p2;
```

Koje su sledeće izjave tačne?

-  **p1 = list1;**
-  **p1 = p2;**
 - **list1 = p1;**
 - **list1 = list2;**
 - **sve od datog**

NIZOVI, STRINGOVI I POKAZIVAČI

34. Analizirati sledeći kod:

```
#include <iostream>
using namespace std;
int main() {
    char t[10];
    char* p = t;
    cout << "Enter a string: ";
    cin >> p;
    cout << p << endl;
    return 0;
}
```

- Ako se pokrene program i unese ulaz kao abc, ništa se neće prikazati na ekranu. Program generiše greške
- Ako se pokrene program i unese ulaz kao abc, prikazaće se abc.
- Ako se pokrene program i unese ulaz kao abc, nepredvidivi znakova će biti prikazani
- Ako se pokrene program i unese ulaz kao abc, pojaviće se greška kompilacije (engl. run time error), jer se p koristio bez inicijalizacije

NIZOVI, STRINGOVI I POKAZIVAČI

34. Analizirati sledeći kod:

```
#include <iostream>
using namespace std;
int main() {
    char t[10];
    char* p = t;
    cout << "Enter a string: ";
    cin >> p;
    cout << p << endl;
    return 0;
}
```

- Ako se pokrene program i unese ulaz kao abc, ništa se neće prikazati na ekranu. Program generiše greške
- 😊 Ako se pokrene program i unese ulaz kao abc, prikazaće se abc.
- Ako se pokrene program i unese ulaz kao abc, nepredvidivi znakova će biti prikazani
- Ako se pokrene program i unese ulaz kao abc, pojaviće se greška kompilacije (engl. run time error), jer se p koristio bez inicijalizacije

NIZOVI, STRINGOVI I POKAZIVAČI

35. Operator new određuje:

- **novu funkciju**
- **novu klasu**
- **dodeljuje memoriju dinamičkom objektu**



NIZOVI, STRINGOVI I POKAZIVAČI

35. Operator new određuje:

- novu funkciju
- novu klasu
- 😊 dodeljuje memoriju dinamičkom objektu



NIZOVI, STRINGOVI I POKAZIVAČI

36. Operator new se koristi za:

- Deklarisanje nove promenljive
- Dinamicku dodelu memorije
- Deklarisanje nove funkcije



NIZOVI, STRINGOVI I POKAZIVAČI

36. Operator new se koristi za:

- Deklarisanje nove promenljive
- 😊 Dinamicku dodelu memorije
- Deklarisanje nove funkcije



NIZOVI, STRINGOVI I POKAZIVAČI

37. Operator delete se koristi za:

- **Brisanje promenljive**
- **Brisanje funkcije**
- **Oslobađanje dinamički dodeljene memorije**



NIZOVI, STRINGOVI I POKAZIVAČI

37. Operator delete se koristi za:

- **Brisanje promenljive**
- **Brisanje funkcije**
- ☺ **Oslobađanje dinamički dodeljene memorije**



NIZOVI, STRINGOVI I POKAZIVAČI

38. Operator _____ dinamički alocira memoriju za objekat zadatog tipa.

- **new**
- **delete**
- **extern**



NIZOVI, STRINGOVI I POKAZIVAČI

38. Operator _____ dinamički alocira memoriju za objekat zadatog tipa.



new

- **delete**
- **extern**



NIZOVI, STRINGOVI I POKAZIVAČI

39. Koje su od sledećih izjava tačne?

- **int* pValue = new double;**
- **int* pValue = new int;**
- **double* pValue = new double;**
- **double* pValue = new int;**



NIZOVI, STRINGOVI I POKAZIVAČI

39. Koje su od sledećih izjava tačne?

- `int* pValue = new double;`
- 😊 `int* pValue = new int;`
- 😊 `double* pValue = new double;`
- `double* pValue = new int;`



NIZOVI, STRINGOVI I POKAZIVAČI

40. Neka je niz list definisan kao:

```
int* list = new int[10];
```

Kako se može uništiti list?

- **delete list;**
- **delete* list;**
- **delete [] list;**
- **delete [] *list;**

NIZOVI, STRINGOVI I POKAZIVAČI

40. Neka je niz list definisan kao:

```
int* list = new int[10];
```

Kako se može uništiti list?

- **delete list;**
- **delete* list;**
- 😊 **delete [] list;**
- **delete [] *list;**

NIZOVI, STRINGOVI I POKAZIVAČI

41. Koja je od sledećih izjava ispravna da izbriše dinamički objekat sa pokazivača p?

- `delete *p;`
- `delete p;`
- `delete [] *p;`
- `delete [] p;`



NIZOVI, STRINGOVI I POKAZIVAČI

41. Koja je od sledećih izjava ispravna da izbriše dinamički objekat sa pokazivača p?

- delete *p;
- 😊 delete p;
- delete [] *p;
- delete [] p;



NIZOVI, STRINGOVI I POKAZIVAČI

42. Indeks prvog elementa niza u jeziku C++ je:

- 0
- 1



NIZOVI, STRINGOVI I POKAZIVAČI

42. Indeks prvog elementa niza u jeziku C++ je:

😊 0
• 1



NIZOVI, STRINGOVI I POKAZIVAČI

43. Jezik C++ proverava granice niza.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

43. Jezik C++ proverava granice niza.

- Tačno



Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

44. Naredba `int broj[10] = {0};`

- Inicijalizuje sve elemente niza na vrednost 0
- Predstavlja prazan niz
- Deklariše samo jedan element u nizu



NIZOVI, STRINGOVI I POKAZIVAČI

44. Naredba `int broj[10] = {0};`



Inicijalizuje sve elemente niza na vrednost 0

- **Predstavlja prazan niz**
- **Deklariše samo jedan element u nizu**



NIZOVI, STRINGOVI I POKAZIVAČI

45. Naredba `cout << niz[2];`

- **Prikazuje vrednost trećeg elementa niza**
- **Prikazuje vrednost drugog elementa niza**
- **Prikazuje vrednost indeksa drugog elementa niza**



NIZOVI, STRINGOVI I POKAZIVAČI

45. Naredba `cout << niz[2];`

- 😊 Prikazuje vrednost trećeg elementa niza
- Prikazuje vrednost drugog elementa niza
 - Prikazuje vrednost indeksa drugog elementa niza



NIZOVI, STRINGOVI I POKAZIVAČI

46. Naredba `cin >> niz[1];`

- Prikazuje vrednost prvog elementa niza
- Prikazuje vrednost drugog elementa niza
- Prikazuje vrednost za drugi element niza sa standardnog ulaza



NIZOVI, STRINGOVI I POKAZIVAČI

46. Naredba `cin >> niz[1];`

- Prikazuje vrednost prvog elementa niza
- Prikazuje vrednost drugog elementa niza
- 😊 Prikazuje vrednost za drugi element niza sa standardnog ulaza



NIZOVI, STRINGOVI I POKAZIVAČI

47. Niz je skup objekata:

- **Različitog tipa**
- **Istog tipa**
- **Mešovitog tipa**



NIZOVI, STRINGOVI I POKAZIVAČI

47. Niz je skup objekata:

- Različitog tipa
- 😊 Istog tipa
- Mešovitog tipa



NIZOVI, STRINGOVI I POKAZIVAČI

48. Naredba for mora se upotrebiti za inicijalizaciju svih elemenata niza na 0.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

48. Naredba for mora se upotrebiti za inicijalizaciju svih elemenata niza na 0.

- Tačno



Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

49. Za sabiranje elemenata dvodimenzionalnog niza moraju se upotrebiti ugnježdene for petlje.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

49. Za sabiranje elemenata dvodimenzionalnog niza moraju se upotrebiti ugnježdene for petlje.



Tačno

- **Pogrešno**



NIZOVI, STRINGOVI I POKAZIVAČI

50. C-string je niz znakova koji:

- Počinje od početka reda
- Završava se znakom `\0`
- Ima najmanje jedan znak



NIZOVI, STRINGOVI I POKAZIVAČI

50. C-string je niz znakova koji:

- Počinje od početka reda
- 😊 Završava se znakom \0
- Ima najmanje jedan znak



NIZOVI, STRINGOVI I POKAZIVAČI

51. Standardna biblioteka funkcija koje rade sa C stringovima je:

- **iostream**
- **cstdlib**
- **cstring**



NIZOVI, STRINGOVI I POKAZIVAČI

51. Standardna biblioteka funkcija koje rade sa C stringovima je:

- **iostream**
- **cstdlib**
- 😊 **cstring**



NIZOVI, STRINGOVI I POKAZIVAČI

52. Funkcija strcmp:

- **Komprimuje niz**
- **Poredi dva stringa**
- **Kompajlira niz**



NIZOVI, STRINGOVI I POKAZIVAČI

52. Funkcija strcmp:

- Komprimuje niz
- 😊 Poredi dva stringa
- Kompajlira niz



NIZOVI, STRINGOVI I POKAZIVAČI

53. Funkcija tolower(ch):

- Ispisuje znak na ekranu malim fontom
- Konvertuje znak u malo slovo
- Konvertuje znak u veliko slovo



NIZOVI, STRINGOVI I POKAZIVAČI

53. Funkcija tolower(ch):

- Ispisuje znak na ekranu malim fontom
- 😊 Konvertuje znak u malo slovo
- Konvertuje znak u veliko slovo



NIZOVI, STRINGOVI I POKAZIVAČI

54. Funkcija toupper(ch):

- Ispisuje znak na ekranu većim fontom
- Konvertuje znak u malo slovo
- Konvertuje znak u veliko slovo



NIZOVI, STRINGOVI I POKAZIVAČI

54. Funkcija toupper(ch):

- Ispisuje znak na ekranu većim fontom
- Konvertuje znak u malo slovo
- 😊 Konvertuje znak u veliko slovo



NIZOVI, STRINGOVI I POKAZIVAČI

55. Kada se ustanovi da su dva stringa jednaka, funkcija strcmp vraća:

- 0
- 1
- Pozitivan broj
- Negativan broj



NIZOVI, STRINGOVI I POKAZIVAČI

55. Kada se ustanovi da su dva stringa jednaka, funkcija strcmp vraća:



0

- 1
- Pozitivan broj
- Negativan broj



NIZOVI, STRINGOVI I POKAZIVAČI

56. Dimenzija niza znakova u koji treba da stane string “univerzitet” je:

- 10
- 11
- 12



NIZOVI, STRINGOVI I POKAZIVAČI

56. Dimenzija niza znakova u koji treba da stane string “univerzitet” je:

- 10
- 11
- 😊 12



NIZOVI, STRINGOVI I POKAZIVAČI

57. Funkcija koja sa tastature učitava niz znakova koji sadrži razmake je:

- **cin**
- **gets**



NIZOVI, STRINGOVI I POKAZIVAČI

57. Funkcija koja sa tastature učitava niz znakova koji sadrži razmake je:

- cin



gets



NIZOVI, STRINGOVI I POKAZIVAČI

58. Funkcija koja vraća dužinu stringa je:

- **strcpy**
- **strlen**
- **strcmp**



NIZOVI, STRINGOVI I POKAZIVAČI

58. Funkcija koja vraća dužinu stringa je:

- strcpy
- 😊 strlen
- strcmp



NIZOVI, STRINGOVI I POKAZIVAČI

59. String konstanta se piše pod navodnicima.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

59. String konstanta se piše pod navodnicima.



Tačno

- **Pogrešno**



NIZOVI, STRINGOVI I POKAZIVAČI

60. Naredba `str[] = "nizZnakova";` je ispravna.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

60. Naredba `str[] = "nizZnakova";` je ispravna.



Tačno

- **Pogrešno**



NIZOVI, STRINGOVI I POKAZIVAČI

61. Pokazivač je promenljiva:

- Čija je vrednost adresa druge promenljive
- Koja pokazuje na funkciju
- Koja pokazuje na kraj programa



NIZOVI, STRINGOVI I POKAZIVAČI

61. Pokazivač je promenljiva:



Čija je vrednost adresa druge promenljive

- **Koja pokazuje na funkciju**
- **Koja pokazuje na kraj programa**



NIZOVI, STRINGOVI I POKAZIVAČI

62. Svakom bajtu u memoriji dodeljuje se jedinstvena adresa.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

62. Svakom bajtu u memoriji dodeljuje se jedinstvena adresa.



Tačno

- **Pogrešno**



NIZOVI, STRINGOVI I POKAZIVAČI

63. Ime niza je pokazivač na:

- Sve članove niza
- Veličinu niza
- Prvi element niza



NIZOVI, STRINGOVI I POKAZIVAČI

63. Ime niza je pokazivač na:

- Sve članove niza
- Veličinu niza
- 😊 Prvi element niza



NIZOVI, STRINGOVI I POKAZIVAČI

64. Adresa na koju pokazuje ime niza može se promeniti.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

64. Adresa na koju pokazuje ime niza može se promeniti.

- Tačno



Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

65. Gde je greška u sledećem delu koda?

```
int *p = &i;  
int i;
```

- U prvom redu
- U drugom redu
- U redosledu deklarisanja promenljivih

NIZOVI, STRINGOVI I POKAZIVAČI

65. Gde je greška u sledećem delu koda?

```
int *p = &i;  
int i;
```

- U prvom redu
- U drugom redu
- 😊 U redosledu deklarisanja promenljivih

NIZOVI, STRINGOVI I POKAZIVAČI

66. Na šta pokazuje pokazivač ako ima vrednost 0?

- Na nultu memorijsku lokaciju
- Na prvi član niza
- Ni na šta



NIZOVI, STRINGOVI I POKAZIVAČI

66. Na šta pokazuje pokazivač ako ima vrednost 0?

- Na nultu memorijsku lokaciju
- Na prvi član niza
- 😊 Ni na šta



NIZOVI, STRINGOVI I POKAZIVAČI

67. Šta radi sledeći deo koda?

```
int a[3] = {1, 2, 3};  
int i;  
for (i = 0; i < 3; ++i)  
    cout << *(a+i);
```

- **Sabira članove niza**
- **Ispisuje vrednosti pokazivača na članove niza**
- **Ispisuje članove niza**



NIZOVI, STRINGOVI I POKAZIVAČI

67. Šta radi sledeći deo koda?

```
int a[3] = {1, 2, 3};  
int i;  
for (i = 0; i < 3; ++i)  
    cout << *(a+i);
```

- **Sabira članove niza**
- **Ispisuje vrednosti pokazivača na članove niza**
- 😊 **Ispisuje članove niza**



NIZOVI, STRINGOVI I POKAZIVAČI

68. Pokazivač može da pokazuje ni na šta.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

68. Pokazivač može da pokazuje ni na šta.



Tačno

- **Pogrešno**



NIZOVI, STRINGOVI I POKAZIVAČI

69. Ako želim da pokazivač `p` pokazuje na promenljivu `x`, naredba kojom ću to postići je:

- `p = x;`
- `p = &x;`
- `p = *x;`
- `*p = x;`



NIZOVI, STRINGOVI I POKAZIVAČI

69. Ako želim da pokazivač `p` pokazuje na promenljivu `x`, naredba kojom ću to postići je:

- `p = x;`
- 😊 `p = &x;`
- `p = *x;`
- `*p = x;`



NIZOVI, STRINGOVI I POKAZIVAČI

70. Šta će se ispisati na ekranu posle sledećih naredbi?

```
int *p;  
int x = 100;  
p = &x;  
cout << (*p);
```

- 0
- 100
- Memorijska adresa promenljive x



NIZOVI, STRINGOVI I POKAZIVAČI

70. Šta će se ispisati na ekranu posle sledećih naredbi?

```
int *p;  
int x = 100;  
p = &x;  
cout << (*p);
```

- 0
- 😊 100
- Memorijska adresa promenljive x



NIZOVI, STRINGOVI I POKAZIVAČI

71. Kolika je vrednost promenljive x nakon sledećih naredbi?

```
int *p, x;  
x = 100;  
p = &x;  
p++;
```

- 100
- 101
- 104



NIZOVI, STRINGOVI I POKAZIVAČI

71. Kolika je vrednost promenljive x nakon sledećih naredbi?

```
int *p, x;  
x = 100;  
p = &x;  
p++;
```

-  100
- 101
 - 104



NIZOVI, STRINGOVI I POKAZIVAČI

72. Kolika je vrednost promenljive x nakon sledećih naredbi?

```
int *p, x;  
x = 100;  
p = &x;  
(*p)++;
```

- **100**
- **101**
- **104**



NIZOVI, STRINGOVI I POKAZIVAČI

72. Kolika je vrednost promenljive x nakon sledećih naredbi?

```
int *p, x;  
x = 100;  
p = &x;  
(*p)++;
```

- 100
- 😊 101
- 104



NIZOVI, STRINGOVI I POKAZIVAČI

73. Moguće je definisati niz pokazivača.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

73. Moguće je definisati niz pokazivača.



Tačno

- **Pogrešno**



NIZOVI, STRINGOVI I POKAZIVAČI

74. U radu sa pokazivačima koriste se operatori:

- **&**
- **.**
- *****
- **&&**



NIZOVI, STRINGOVI I POKAZIVAČI

74. U radu sa pokazivačima koriste se operatori:



&

•

.



*

•

&&



NIZOVI, STRINGOVI I POKAZIVAČI

75. Jedini ceo broj koji se može direktno dodeliti pokazivaču je 0.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

75. Jedini ceo broj koji se može direktno dodeliti pokazivaču je 0.



Tačno

- **Pogrešno**



NIZOVI, STRINGOVI I POKAZIVAČI

76. Povratna vrednost operatora new je:

- **Pokazivač**
- **void**
- **int**
- **Ništa od navedenog**



NIZOVI, STRINGOVI I POKAZIVAČI

76. Povratna vrednost operatora new je:



Pokazivač

- **void**
- **int**
- **Ništa od navedenog**



NIZOVI, STRINGOVI I POKAZIVAČI

77. Operator delete oslobađa memoriju koju je prethodno rezervisao operator new.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

77. Operator delete oslobađa memoriju koju je prethodno rezervisao operator new.



Tačno

- **Pogrešno**



NIZOVI, STRINGOVI I POKAZIVAČI

78. Nemoguće je kreirati promenljive tokom izvršavanja programa.

- Tačno
- Pogrešno



NIZOVI, STRINGOVI I POKAZIVAČI

78. Nemoguće je kreirati promenljive tokom izvršavanja programa.

- Tačno



Pogrešno





KRAJ!

