



Kolokviumi i parë në lëndën “Algoritmet dhe Strukturat e të Dhënave”, 30.04.2024

Emri Mbiemri:	ID:	Gr. ushtrimeve:
---------------	-----	-----------------

1. (3p) Të tregohet dalja në ekran pasi të ekzekutohet programi në vijim. Rezultatet i shënoni në fletore. Secili rresht i daljes ka 0.5 pikë.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int a = 3, b = 1, c = -1;
6     int *pa = &a, *pb = &b, *pc = &c;
7     int *pv[3] = { pa, pb, pc };
8     int **px = pv + 1;
9
10    cout << "1: " << *pb - *pc << endl;
11    cout << "2: " << 3 * *pa << endl;
12    cout << "3: " << *pv[1] + 1 << endl;
13    cout << "4: " << *(pv + *pb) << endl;
14    cout << "5: " << *(&(pv[1]) - 1) << endl;
15    cout << "6: " << *(px[1]) << endl;
16    return 0;
17 }
```

2. (3p) Çfarë do të shfaqet në ekran kur të ekzekutohet kodi në vijim? (Rezultatit shënojeni në fletore.)

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

int main() {
    const int n = 6;
    int v[n] = { 5, -3, 4, 7, -1, 6 };
    stack<int> steku = stack<int>();
    for (int i = 0; i < n; i++) {
        steku.push(-1);
        if (v[i] > 0) {
            steku.push(v[i]);
        }
    }
    while (!steku.empty()) {
        cout << steku.top() << " ";
        steku.pop();
    }
    return 0;
}
```

3. (3p) Të deklarohet klasa **Drejtkenedeshi** me anëtarët në vijim.

- (1p) Fushat private **a** dhe **b** (numra të plotë).
- (1p) Konstruktoren publik që inicializon fushat private përmes parametrave me emra të njëjtë.
- (1p) Metodatat publike **merrA()** dhe **merrB()** të cilat kthejnë vlerat e fushave përkatëse.

4. (3p) Të shkruhet (jashtë klasës) funksioni **llogarit** i cili pranon tre parametra: një parametër përmes referencës të tipit **Drejtkenedeshi** (nga detyra paraprake), dhe dy parametra tjerë përmes referencës **s** dhe **p** si numra të plotë. Funksioni njehëson sipërfaqën dhe perimetrin e drejtkëndëshit dhe rezultatet i vendos në parametrat përkatës **s** dhe **p**. Funksioni nuk kthen asgjë.

Në **main** të deklarohet një drejtkëndësh me vlera sipas dëshirës dhe të llogaritet sipërfaqja dhe perimetri i atij drejtkëndëshi përmes një thirrjeje të vetme të funksionit **llogarit**. Rezultatet e fituara të shtypen në ekran.

5. (4p) Të shkruhet programi i cili lexon dhe përpunon pikët e provimeve sipas përshkrimit në vijim.

- (1p) Programi vazhdimisht lexon numra të plotë nga tastiera, deri kur të lexohet një numër negativ.
- (1p) Pikët e lexuara ruhen në një listë të tipit **vector<int>** (pa përfshirë numrin e fundit negativ).
- (1p) Të llogaritet mesatarja e pikëve dhe të shtypet në ekran.
- (1p) Të trajtohet rasti nëse lista është e zbrazët, me ç'rast nuk shtypen pikët, por mesazhi “Nuk ka vlera.”.

6. (3p) Përmes notacionit Big-O, të tregohet rasti më i mirë, mesatar, dhe ai më i keq për problemet në vazhdim. Për secilën kërkesë, tregoni nga një shembull kur ndodh rasti më i mirë dhe ai më i keq. Rezultatet i shënoni në fletore.

- (1p) A gjendet ndonjë element negativ në varg?
- (1p) Largimi i elementit nga fillimi i array listës.
- (1p) A është n numër i thjeshtë (nuk plotëpjestohet me $2 \dots n - 1$)?



UNIVERSITETI I PRISHTINËS "HASAN PRISHTINA"
FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE

Bregu i Diellit, 10 000 Prishtinë, Republika e Kosovës
Tel: +383 38 554 896 ext.102 · Email: fiek@uni-pr.edu · www.uni-pr.edu

7. (3p) Çfarë do të shfaqet në ekran kur të ekzekutohet kodi në vijim? (Rezultatit shënojeni në fletore.)

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 class Student {
6 public:
7     string emri;
8     double notaMesatare;
9
10    Student(string emri, double notaMesatare) {
11        this->emri = emri;
12        this->notaMesatare = notaMesatare;
13    }
14 };
15
16 Student* kerko(vector<Student*> &v, double nota){
17     for (int i = 0; i < v.size(); i++) {
18         if (v[i]->notaMesatare >= nota) {
19             return v[i];
20         }
21     }
22     return NULL;
23 }
24
25 void shtyp_emrin(Student *studenti) {
26     cout << (studenti != NULL ? studenti->emri : "Nuk ekziston") << endl;
27 }
28
29 int main()
30 {
31     auto studentet = vector<Student*>();
32     studentet.push_back(new Student("Anda", 8.6));
33     studentet.push_back(new Student("Blerona", 9.1));
34     studentet.push_back(new Student("Clirim", 9.4));
35     studentet.push_back(new Student("Dren", 7.5));
36     shtyp_emrin( kerko(studentet, 9.0) );
37     shtyp_emrin( kerko(studentet, 9.5) );
38     shtyp_emrin( kerko(studentet, 7.5) );
39     return 0;
40 }
```

8. (3p) Të shkruhet funksioni **transfero<T>(&dst, &src)** i cili bart të gjitha elementet nga radha (queue) **src** në fundin e radhës **dst**. Funksioni nuk kthen asgjë dhe punon me çfarëdo tipi **T** (kujtesë: **template<typename T>**).

Për shembull, nëse kemi radhët $q_1 = \underline{1\ 2\ 3}$ dhe $q_2 = \underline{4\ 5\ 6}$, pas thirrjes së **transfero(q_1 , q_2)**, do të përfundojmë me $q_1 = \underline{1\ 2\ 3\ 4\ 5\ 6}$ dhe $q_2 = \emptyset$.

Referenca e librarisë standarde

Veprimi	vector<T>	stack<T> (LIFO)	queue<T> (FIFO)
Shtimi i elementeve	push_front(x), push_back(x)	push(x)	push(x)
Largimi i elementeve	pop_front(), pop_back()	pop()	pop()
Leximi i elementeve	v[i]	top()	front(), back()
Nr. i elementeve	size()	size()	size()
E zbrazët?	empty()	empty()	empty()