სახელი: ქულა:

ფინალური გამოცდა

პროგრამირებაის აბსტრაქციები

5 ივლისი, 10:00 – 12:45

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1  20 ქულა | 2  20 ქულა | 3  25 ქულა | 4  35 ქულა | 5  35 ქულა | 6  40 ქულა |
|  |  |  |  |  |  |

**ამოცანა 1.** (20 ქულა) ბეჭდურა

დაწერეთ რას დაბეჭდავს ქვემოთ მოცემული პროგრამა:

ა)

int a(int &k, int \* p) {

\*p = 7;

k = \*p + 3;

return 0;

}

int main()

{

int k;

int \* p = new int;

cout << k << endl;

cout << \*p << endl;

a(k, p);

cout << k << endl;

cout << \*p << endl;

return (0);

}

პასუხი:

ბ)

int a(int &k, int \* p) {

\*p = 7;

k = \*p + 3;

return 0;

}

int main()

{

int \* p = new int[5];

\*p = 3;

p[1] = p[0] + 3;

p[2] = p[1] + p[0];

p[3] = p[2] + \*p;

cout << \*p << endl;

for (int i=0; i<5; i++) {

cout << p[i] << endl;

}

return (0);

}

პასუხი:

**ამოცანა 2.(20 ქულა) ბმულის სიის სიმრავლე**

struct Node {

string name;

Node \* next;

};

Node სტრუქტურის საშუალებით მოცემული გაქვთ ბმულის სია. თქვენი ამოცანაა დაწეროთ removeDublicates ფუნქცია, რომელიც გადაეცემული სიიდან შლის დუბლიკატ წევრებს. ბმული სიის ორი წევრი ითვლება დუბლიკატად თუ მათი name მნიშვლელობები ერთი და იგივეა.

void removeDublicates(Node\* head){

**ამოცანა 3.(25 ქულა) ექპერიმენტები ენზე**

მოცემული გაქვთ მთელი რიცხვების მასივი a და მთელი რიცხვი n. n რიცხვზე შეგვიძლია ჩავატაროთ შემდეგი ოპერაციები: თუკი n იყოფა a[i]-ზე (მასივის რომელიმე წევრზე) მაშინ n-იდან შეგვიძლია მივიღოთ რიცხვი n/a[i]. ასევე n-იდან შეგვიძლია მივიღოთ რიცხვი (n-1). ოპერაციებს ვწყვეტთ როცა n=1. თქვენი მიზანია დაითვალოთ რამდენი განსხვავებული გზით შეიძლება მივიდეთ 1-მდე.

მაგალითად თუ n=3, a={2,3} მაშინ  
3->(n-1)->2->(n-1)->1  
3->(n-1)->2->(n/2)->1

3->(n/3)->1

სულ მივიღეთ 3 განსხვავებული გზა.

**int PathCount(int n, int[]a){**

**ამოცანა 4. (35 ქულა) საერთო მშობელი.**

მოცემული გაქვთ ორობითი ხე და ამ ხეში მდებარე ორ წვეროზე მიმთითებელი. თქვენი მიზანია იპოვოთ ამ ორი წვეროს უახლოესი საერთო მშობელი.

struct Node{

Node \* left, \*right;

//some values  
};

Node \* ParentSearch(Node\* root, Node\* childOne, Node\* childTwo){

**ამოცანა 5. (35 ქულა) გრაფი მონტე კრისტო**

მოცემული გაქვთ არამიმართული ბმული გრაფი. რა მაქსიმალური რაოდენობის წიბოს წაშლა არის შესაძლებელი იმისათვის რომ გრაფი კვლავ ბმული დარჩეს? თქვენი ამოცანაა დაწეროთ მეთოდი DeleteArcs რომელიც დააბრუნებს წასაშლელი წიბოების ვექტორს.

**struct Node;**

**struct Arc {**

**Node \* start;**

**Node \* finish;**

**};**

**struct Node {**

**Vector<Arc> ways;**

**};**

**Vector<Node \*> DeleteArcs(Vector<Node \*> graph){**

**ამოცანა 6. (40 ქულა) ფაილური სისტემა**

FileSystem კლასი აღწერს ფაილური სისტემის სტრუქტურას.

კლასს აქვს შემდეგი ფუნქციები:

Vector<string> getFolderList(); - აბრუნებს აქტიურ ფოლდერში არსებული ფოლდერების სიას.

Vector<string> getFileList(); - აბრუნებს აქტიურ ფოლდერში არსებული ფაილების სიას.

bool createFolder(string name); - აქტიურ ფოლდერში ქმნის name სახელის ფოლდერს. აბრუნებს true ს თუ ასეთი ფოლდერი არ არსებობდა და false ს თუ ფოლდერი უკვე არსებობდა.

bool createFile(string name); - აქტიურ ფოლდერში ქმნის name სახელის ფაილს. აბრუნებს true ს თუ ასეთი ფაილი არ არსებობდა და false ს თუ ფაილი უკვე არსებობდა.

bool selectFolder(string name); - გააქტიურებს აქტიურ ფოლდერში არსებულ name სახელის მქონე ფოლდერს. აბრუნებს true ს თუ ასეთი ფოლდერი არსებობს და false ს თუ ფოლდერი არ არსებობს.

თქვენი მიზანია მოიფიქროთ სტუქტურა რითაც აღიწერება ეს კლასი და გააკეტოთ ამ კლასის ხუთივე მეთოდის იმპლემენტაცია.