Структурно Програмирање – задачи за вежбање

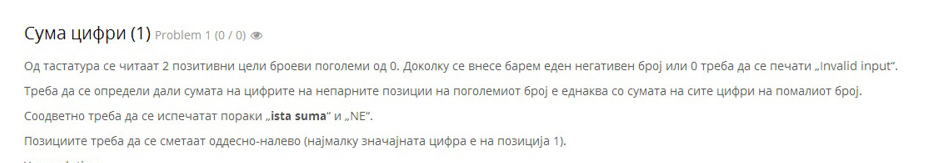
#include <iostream>  
using namespace std;  
int main() {  
 int n,m;  
 cin>>n>>m;  
 if(m<=0 || n<=0) {  
 cout<<"Invalid input";  
 }else {  
 int max;  
 int min;  
 if(m>n) {  
 max=m;  
 min=n;  
 }else {  
 max=n;  
 min=m;  
 }  
 int pozicija=1;  
 int novbr=1;  
 int filtered=0;  
 while(max>0) {  
 int cifra = max%10;  
 if(pozicija%2!=0) {  
 filtered += cifra \* novbr;  
 novbr \*= 10;  
 }  
 max/=10;  
 pozicija++;  
 }  
 if (filtered == min)  
 cout << "DA" << endl;  
 else  
 cout << "NE" << endl;  
  
 }  
 return 0;  
}



//ne e celosna  
#include <iostream>  
using namespace std;  
  
int main() {  
 int broj;  
 int rezultat = 0;  
 int najmal = 1000000; // доволно голем број  
 int najgolem = -1;  
 while(cin>>broj) {  
 int temp = broj;  
 int pozicija = 1;  
 int sumParniPozicii= 0;  
 int sumNeparniPozicii = 0;  
 while(temp>0) {  
 int cifra = temp%10;  
 if(pozicija%2==0) {  
 sumParniPozicii += cifra;  
 }else  
 sumNeparniPozicii += cifra;  
 temp/=10; //se trga poslednata cifra  
 pozicija++; //se menuva brojot na pozicijata za 1  
 //ciklusot prodolzhuva da vrti dodeka temp(broj) ne stane 0!  
 }  
 cout<<"zbirot na broevi na parni pozicii "<<sumParniPozicii<<endl;  
 cout<<"zbirot na broevi na neparni pozicii "<<sumNeparniPozicii<<endl;  
 }  
 return 0;  
}



#include <iostream>  
using namespace std;  
  
int main() {  
 int m,n;  
 if(m<=0 || n<=0) { // && - prifakja i edno tochno edno netochno  
 // || - mora i dvata uslovi da se tochni  
 cout<<"Invalid input"<<endl;  
 }  
 int pogolem,pomal;  
 if(m>n) {  
 pogolem=m;  
 }else  
 pomal=n;  
  
   
  
 return 0;  
}



#include <iostream>

using namespace std;

int main() {

int a, b;

cin >> a >> b;

if (a <= 0 || b <= 0) {

cout << "Invalid input" << endl;

return 0;

}

int pogolem = (a > b) ? a : b;

int pomal = (a < b) ? a : b;

// Сума на непарни позиции од поголемиот број

int pozicija = 1;

int sumaNeparni = 0;

while (pogolem > 0) {

int cifra = pogolem % 10;

if (pozicija % 2 != 0) {

sumaNeparni += cifra;

}

pogolem /= 10;

pozicija++;

}

// Сума на сите цифри од помалиот број

int sumaSite = 0;

while (pomal > 0) {

sumaSite += pomal % 10;

pomal /= 10;

}

if (sumaNeparni == sumaSite)

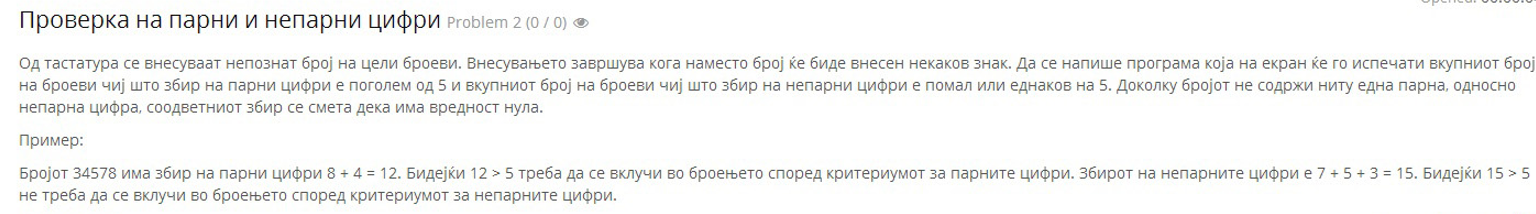
cout << "ista suma" << endl;

else

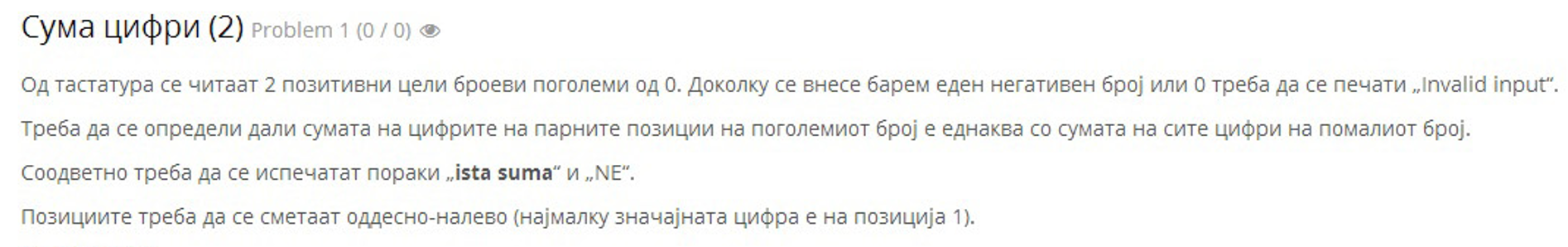
cout << "NE" << endl;

return 0;

}



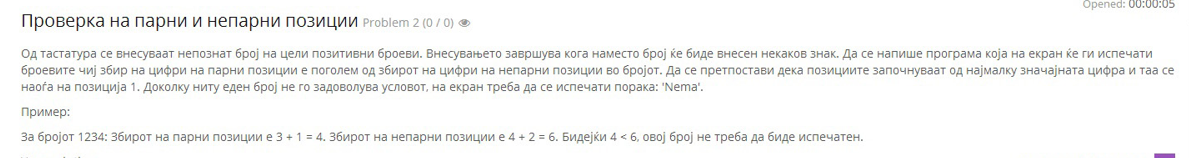
#include <iostream>  
using namespace std;  
  
int main() {  
 int broj;  
 int sumParni=0, sumNeparni=0;  
 int brojachParni=0, brojachNeparni=0;  
 int pomali5neparni=0;  
 while (cin >> broj) {  
 int temp = broj;  
 //ciklus za parni  
 while(temp!=0) {  
 int cifra = temp%10;  
 if(cifra%2==0) {  
 sumParni+=cifra; //suma od parnite broevi  
 brojachParni++;  
 }else{  
 sumNeparni+=cifra;  
 brojachNeparni++;  
 }  
 if(sumNeparni<=5) {  
 pomali5neparni++;  
 }  
 temp/=10;  
 }  
  
 }  
cout<<"Broevi so zbir pogolem od 5 "<<brojachParni<<endl;  
 cout<<"Broevi so zbir pomal ili ednakov od 5 "<<pomali5neparni<<endl;  
 return 0;  
}  
  
/\*  
#include <iostream>  
using namespace std;  
  
int main() {  
int broj;  
int sumParni = 0, sumNeparni = 0;  
int brojachParni = 0, brojachNeparni = 0;  
int pomali5neparni = 0;  
  
while (cin >> broj) {  
int temp = broj;  
int localSumParni = 0, localSumNeparni = 0;  
  
while (temp != 0) {  
int cifra = temp % 10;  
if (cifra % 2 == 0) {  
localSumParni += cifra;  
} else {  
localSumNeparni += cifra;  
}  
temp /= 10;  
}  
  
if (localSumParni > 5) {  
brojachParni++;  
}  
if (localSumNeparni <= 5) {  
pomali5neparni++;  
}  
  
// Optional: track total sum across all numbers  
sumParni += localSumParni;  
sumNeparni += localSumNeparni;  
}  
  
cout << "Broevi so zbir pogolem od 5: " << brojachParni << endl;  
cout << "Broevi so zbir pomal ili ednakov od 5: " << pomali5neparni << endl;  
  
return 0;  
}  
\*/

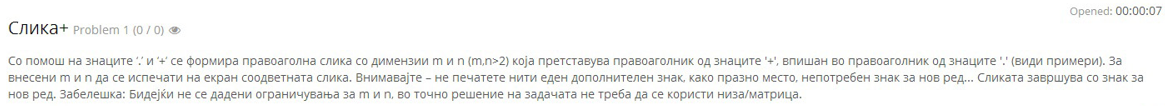


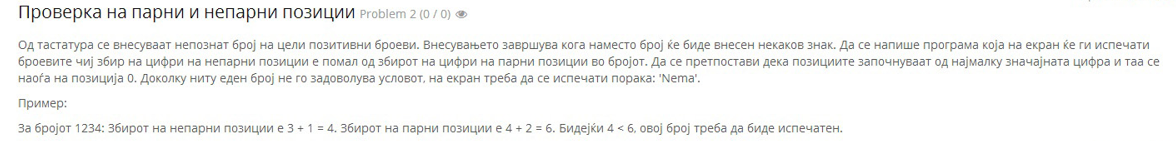
#include <iostream>  
using namespace std;  
//reshena so AI  
  
int main() {  
 int a, b;  
 cin >> a >> b;  
  
 if (a <= 0 || b <= 0) {  
 cout << "Invalid input";  
 return 0;  
 }  
  
 int pogolem, pomal;  
 if (a > b) {  
 pogolem = a;  
 pomal = b;  
 } else {  
 pogolem = b;  
 pomal = a;  
 }  
  
 cout << "pogolem e " << pogolem << " , a pomal e " << pomal << endl;  
  
 // Suma na cifri od parni pozicii vo pogolemiot broj  
 int pozicija = 1;  
 int tempPogolem = pogolem;  
 int sumParniPozicii = 0;  
  
 while (tempPogolem != 0) {  
 int cifra = tempPogolem % 10;  
 if (pozicija % 2 == 0) {  
 sumParniPozicii += cifra;  
 }  
 tempPogolem /= 10;  
 pozicija++;  
 }  
  
 // Suma na site cifri od pomaliot broj  
 int tempPomal = pomal;  
 int sumSiteCifri = 0;  
  
 while (tempPomal != 0) {  
 int cifra = tempPomal % 10;  
 sumSiteCifri += cifra;  
 tempPomal /= 10;  
 }  
  
 // Proverka  
 if (sumParniPozicii == sumSiteCifri) {  
 cout << "ista suma" << endl;  
 } else {  
 cout << "NE" << endl;  
 }  
  
 return 0;  
}

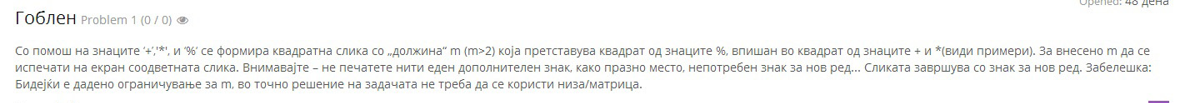








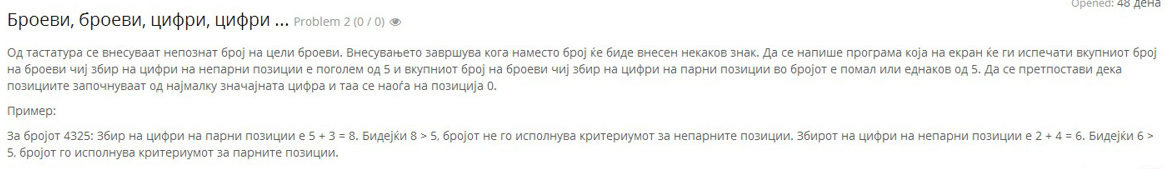




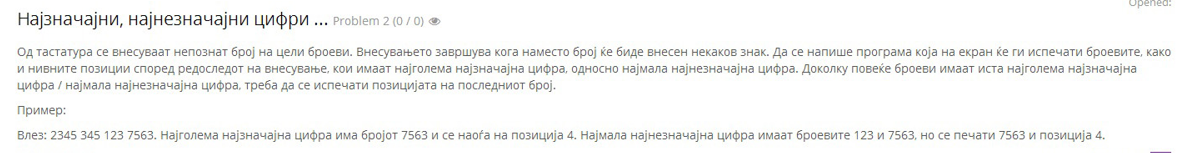


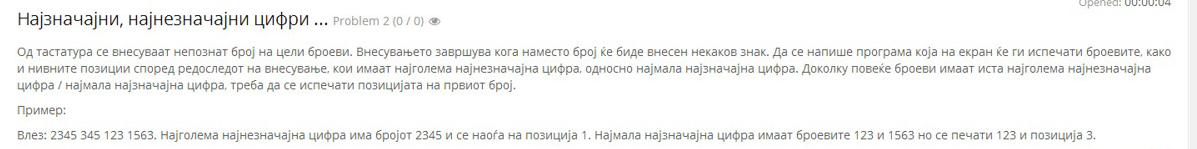


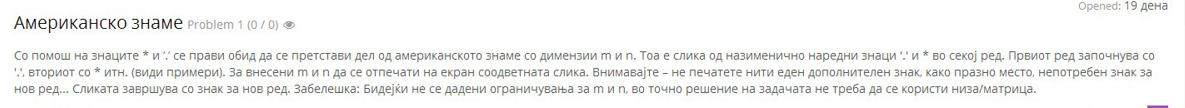


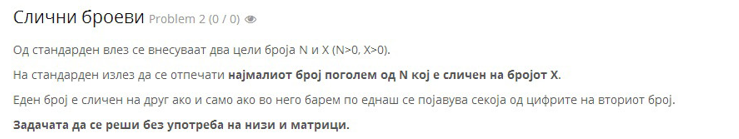


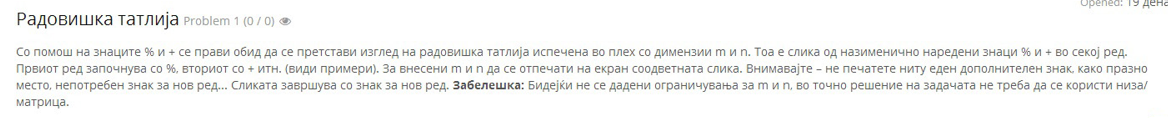


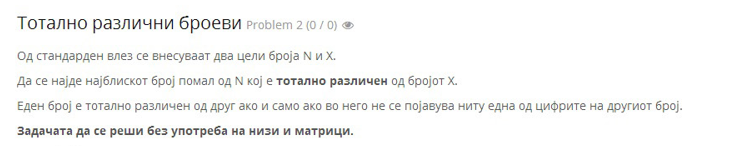




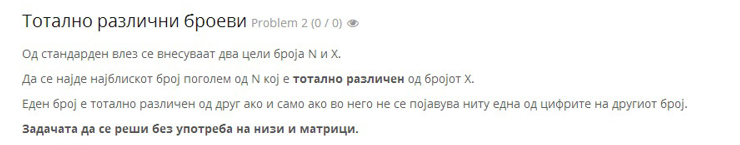


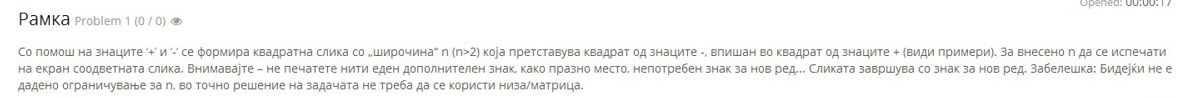


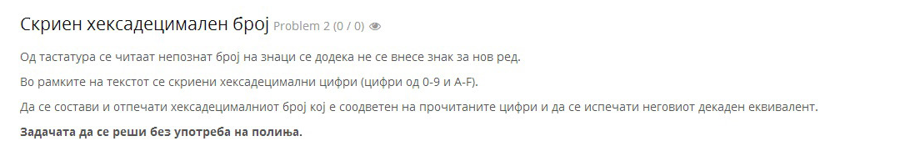




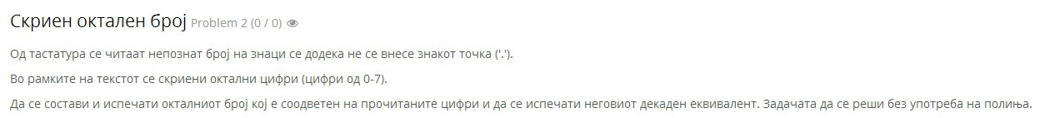




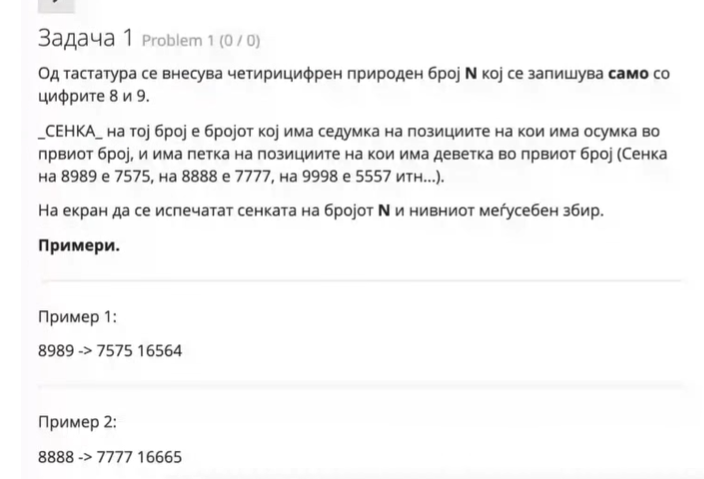








Aud vezhbi Stefan andonov



Zadacha: Обиколка на геометриска слика се пресметува како збир од должините од сите нејзини страни. Во оваа задача, на почеток како влез се внесува една буква Т или М. Во зависност од тоа: Ако е внесена буквата Т, тоа означува дека внесуваме податоци за триаголник и се внесуваат 3 различни цели броеви (должини на неговите страни) Ако е внесена буквата М, тоа означува дека внесуваме податоци за многуаголник и се внесува прво број N - број на страните на многуаголникот, а потоа N различни цели броеви (должини на неговите страни). Ваша задача е на излез да ја испечатите обиколката на внесената геометриска слика. Забелешка: Секогаш на влез ќе има валидни податоци.

Влез: Т 6 7 9 Излез: 22 Пример 2: Влез: М 5 3 3 3 4 4 Излез: 17

Задача 2 Problem 2 (0 / 0) Од тастатура се внесуваат два едноцифрени броја C1 и C2 (1 <= C1, C2 <= 9). Потоа се внесува природен број N, кој означува колку броеви ќе се внесат во продолжение. Потоа се внесуваат N-те природни броеви. За секој од внесените броеви програмата треба да го испечати бројот кај кој цифрата C1 од внесениот број е заменетa со цифра C2, како и неговиот следбеник. Забелешка: За освојување на 80% од поените може да претпоставите дека внесените броеви не завршуваат на нула.

5 6 7 125 521 1556 5015 950 555 5

Zadacha: Од тастатура се вчитува природен број N, а потоа се вчитуваат последователно N тројки цели броеви (a, b, c). Нека за секоја тројка цели броеви a, b и c се дефинира вредност s што се пресметува по следната формула: s = max/2 + 3\*max - max^2 каде што max е најголемиот од броевите a, b, c. На излез да се испечати вредноста s за секоја од тројките броеви и на крај повторно најголемата добиена вредност за s.

Zadacha: Од тастатура се чита позитивен цел број X(>1). Да се најдат сите комбинации од парови ненегативни броеви чиј збир е бројот X. Притоа, за секој пар треба да важи дека првиот број во парот е прост, а вториот е парен број. При печатењето, паровите се печатат во растечки редослед, според првиот елемент од парот. Забелешка: Бројот 1 не е ниту прост, ниту сложен број.

Влез: 17 Излез: 3 + 14 = 17 5 + 12 = 17 7 + 10 = 17 11 + 6 = 17 13 + 4 = 17 17 + 0 = 17